In-Memory Multi-Bit Multiplication and Accumulation

Dissertation submitted to

Visvesvaraya National Institute of Technology, Nagpur

In partial fulfillment of the requirements for the award of the degree

Master of Technology

in
VLSI Design

by

Manasa K

(MT23MVD018)

Under the guidance of

Dr.R.B. Deshmukh

Centre for VLSI and Nanotechnology

Visvesvaraya National Institute of Technology

Nagpur 440010 (India)

June 2025

In-Memory Multi-Bit Multiplication and Accumulation

Dissertation submitted to

Visvesvaraya National Institute of Technology, Nagpur

In partial fulfillment of the requirements for the award of the degree

Master of Technology

in
VLSI Design

by

Manasa K

(MT23MVD018)

Under the guidance of

Dr.R.B. Deshmukh

Centre for VLSI and Nanotechnology

Visvesvaraya National Institute of Technology

Nagpur 440010 (India)

June 2025

© Visvesvaraya National Institute of Technology, Nagpur 2025

Centre for VLSI and Nanotechnology
Visvesvaraya National Institute of Technology, Nagpur

DECLARATION

I, **Manasa K**, hereby declare that this dissertation titled **"In-Memory Multi-Bit Multiplication and Accumulation"** is carried out by me in the **Centre for VLSI and Nanotechnology** of VNIT, Nagpur. The work is original and has not been submitted earlier in whole or part for the award of any degree/diploma.

Manasa K

Date: _____

CERTIFICATE

This is to certify that the dissertation titled **"In-Memory Multi-Bit Multiplication and Accumulation "** is submitted by **Manasa K (MT23MVD018)** in partial fulfillment of the requirements for the award of **Master of Technology in VLSI Design**, VNIT Nagpur. The work is comprehensive, complete and fit for final evaluation.

Dr.R.B. Deshmukh
Centre for VLSI and Nanotechnology
VNIT, Nagpur

Prof. Sanjay B. Dhok
Head, Centre for VLSI and Nanotechnology
VNIT, Nagpur
Date: _____

Centre for VLSI and Nanotechnology

Visvesvaraya National Institute of Technology, Nagpur

June 2025

Date: _____

Non-Plagiarism Certificate

Certified that M. Tech Dissertation titled **"In-Memory Multi-Bit Multiplication and Accumulation"** submitted by **Manasa K (MT23MVD018)** has been checked for plagiarism using TURNITIN software and the overlap is within prescribed limits.

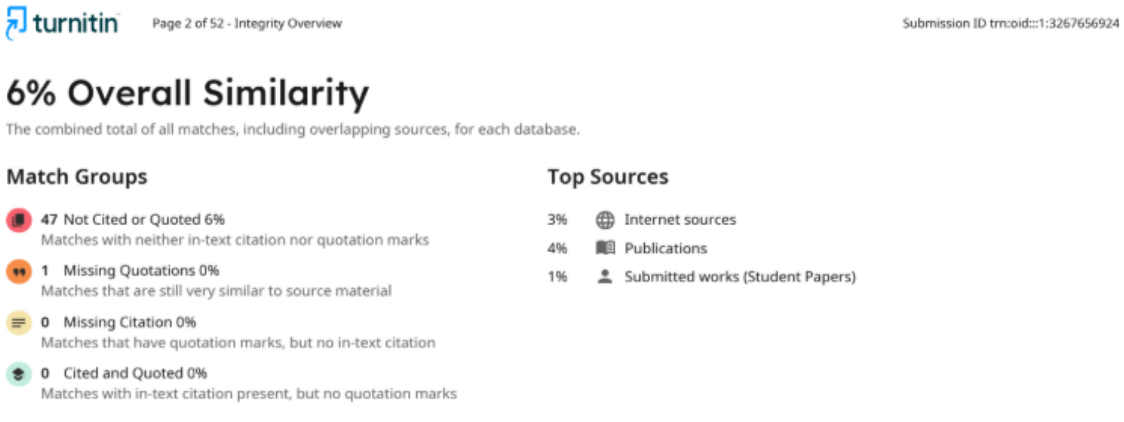
turnitin
Page 2 of 52 - Integrity Overview
Submission ID trn:oid:::1:3267656924
6% Overall Similarity
The combined total of all matches, including overlapping sources, for each database.
Match Groups
47 Not Cited or Quoted 6%
Matches with neither in-text citation nor quotation marks
1 Missing Quotations 0%
Matches that are still very similar to source material
0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation
0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks
Top Sources
3% Internet sources
4% Publications
1% Submitted works (Student Papers)

Figure A: Plagiarism Certificate from Turnitin Software

Manasa K

Date: _____

Dr.R.B. Deshmukh

Supervisor

Acknowledgment

It gives me immense pleasure to express my deep gratitude and heartfelt appreciation to all those who have supported and guided me during the course of my M.Tech project.

I express my gratitude, regards and heartfelt respect to **Dr.R.B. Deshmukh** for his endless and valuable guidance prior to and during the tenure of the project work. His enormous knowledge and intelligence always helped me unconditionally to solve various problems. It has been a rewarding experience working under his supervision .

I would like to acknowledge Head of Department **Prof. S.B. Dhok, Prof.Patrikar , Dr. Vasu Pulijala, Dr. G.C Patil and Dr Nikhil Deep Gupta** of the department, for their support and advice that improved my ways of approach that contributed to the completion of this project.

I extend my sincere gratitude to the institute's leadership for their unwavering support of student academic requirements. My thanks go to **Prof. P. L. Patel**, Director; **Prof. V. R. Kalamkar**, Dean (Academics); and **Prof. R. V. Taiwade**, Associate Dean (Academics - PG/PhD).

I am especially thankful to my parents and family members for their unwavering love, constant encouragement, and sacrifices that have always motivated me to strive for excellence.

My sincere thanks also extend to my batch mates, lab colleagues, and friends for their collaborative spirit and moral support throughout this journey. Their presence made this experience intellectually enriching and personally fulfilling.

Manasa K

Abstract

With the growing need for high-performance and energy-efficient computing systems, particularly in domains such as artificial intelligence, machine learning, and big data analytics, the limitations of conventional von Neumann architectures have become increasingly evident. One of the primary challenges is the significant overhead caused by continuous data exchange between memory and processing units, commonly known as the "memory wall." In-memory computing (IMC) has emerged as an innovative solution to this bottleneck by facilitating computation directly within the memory array, thereby minimizing the energy and time costs associated with data transfer.

This thesis focuses on the design and realization of a multiply-accumulate (MAC) unit using an in-memory computing framework. The proposed architecture leverages modified memory cells that perform dual roles—data storage and computation enabling fundamental arithmetic operations to be executed within the memory itself. By embedding computation into the memory structure, the MAC unit achieves enhanced performance in terms of silicon area, power efficiency, and operational speed when compared to traditional processing approaches.

Among the fundamental operations in AI and ML workloads, Multiply-Accumulate (MAC) operations are the most frequent and computationally expensive. They serve as the backbone of various algorithms, including convolutional neural networks (CNNs), matrix multiplications, and signal processing tasks. Optimizing MAC operations can lead to substantial improvements in overall system performance.

The architecture is modeled and evaluated using standard VLSI design tools, and key performance indicators such as area utilization, power consumption, and delay are thoroughly analyzed. Simulation results validate that the IMC-based MAC design provides substantial advantages over conventional systems, highlighting its suitability for data-intensive and energy-sensitive applications. This work illustrates the promising potential of in-memory computing as a scalable and efficient paradigm for future-generation computing platforms.

List of Figures

1.1 Von-Neumann Arch. 2
1.2 In-memory computing Arch. 3

3.1 Block diagram of proposed architecture 13

4.1 Block diagram of Decoder 18
4.2 Schematic of Decoder 18
4.3 Block diagram of SRAM 9T1C 19
4.4 Schematic of SRAM 6T 20
4.5 Schematic of Write driver 20
4.6 Schematic of Precharge circuit 21
4.7 Schematic of SRAM 9T1C cell 21
4.8 Block diagram of praposed C-DAC 22
4.9 Schematic of Binary weighted C-DAC 22
4.10 Schematic of Operational amplifier 22
4.11 Block diagram of SAR-ADC 23
4.12 Schematic of sample and hold circuit 23
4.13 Schematic of Comparator 24
4.14 Schematic of Integrated Central logic and SAR Register 24
4.15 Schematic of 7 bit SAR-ADC 25
4.16 Schematic of IMAC 4-bit by 4-bit multiplication 25
4.17 Schematic of IMAC 4x4 array 26

5.1 Timing Diagram of 2-Bit Address Decoder 27
5.2 Transient Response of 9T 1C SRAM Cell during Read/Write Operation . 28
5.3 Voltage Transfer Curve (VTC) of 9T 1C SRAM Cell 28
5.4 Power Dissipation of 9T1C SRAM Cell 29
5.5 Time-Domain Output of Binary Weighted C-DAC 29
5.6 Schematic of operational amplifier 30
5.7 Gain response of operational amplifier 30
5.8 Power consumption waveform of C-DAC 31
5.9 Transient Response of Sample-and-Hold Circuit 31
5.10 output wave form of comparator 32

5.11 output wave form of SAR logic 32
5.12 output wave form of SAR logic 33
5.13 Overall Operation Waveforms of the MAC Unit 34
5.14 zoomed-in view of MAC output and digital bus 34
5.15 power consumption of IMAC 35
5.16 Transient analysis waveforms of IMAC Array 35
5.17 Total power consumption of IMAC Array 35

List of Tables

2.1 Comparison of Analog IMC and Digital IMC Features 8

3.1 System Specifications 14

5.1 Prelayout simulation results of this work 37

5.2 Benefits and limitations of various components in the proposed IMC architecture 37

Contents

Acknowledgment **i**

Abstract **ii**

List of Figures **iv**

List of Tables **v**

Index of Terms **ix**

1 Introduction **1**

1.1 Background 2
1.2 Motivation 3
1.3 Problem Statement 4
1.4 Objective of the Work 4
1.5 Organisation of the Thesis 4

2 Literature Review **6**

2.1 Evolution of IMC and Architectural Shifts 6
2.1.1 Digital In-Memory Computing 6
2.1.2 Analog In-Memory Computing 7
2.2 Comparative Analysis of Digital IMC & Analog IMC 8
2.3 Existing IMC architectures 8
2.3.1 SRAM-Based In-Memory Computing 8
2.3.2 DRAM-Based In-Memory Computing 9
2.3.3 RRAM-Based In-Memory Computing 9
2.3.4 PCM-Based In-Memory Computing 9
2.3.5 MRAM-Based In-Memory Computing 10
2.3.6 Flash-Based In-Memory Computing 10
2.3.7 Capacitor-Based In-Memory Computing 10
2.4 Research gap identification 11

3 Proposed Methodology **13**

3.1 Block diagram 13

3.2 Design Specifications 14
3.3 Choice of technology 14
3.4 Operation Principle 14
3.4.1 Word line Activation via Address Decoder 14
3.4.2 Input Application through 4-bit C-DACs 15
3.4.3 In-Memory Multiplication in 9T1C SRAM Bitcells 15
3.4.4 Charge-Based Summation via Capacitor Network 15
3.4.5 Analog-to-Digital Conversion using SAR-ADC 15
3.5 Component Level Explanation 16

4 Design and Implementation **17**
4.1 Tool flow and EDA Tool used 17
4.1.1 Schematic Design (Cadence Virtuoso Schematic Editor) 17
4.1.2 Symbol Creation and Hierarchical Integration 17
4.2 Functional Simulation and Analysis (Cadence - ADEL) 17
4.3 Schematic Design and Simulations of all circuit level blocks 18
4.3.1 Address Decoder 18
4.3.2 SRAM 9T1C Bitcell 19
4.3.3 Binary weighted C-DAC 21
4.4 Successive approximation Register-ADC 23
4.5 IMAC 4-bit by 4-bit multiplication 25
4.6 IMAC 4X4 Array 26

5 Results and Discussions **27**
5.1 Decoder Output Waveform and Analysis 27
5.2 SRAM 9T1C Output Waveforms and Analysis 28
5.3 4-bit Binary Weighted Capacitive DAC 29
5.3.1 Operational amplifier 30
5.4 7 bit SAR ADC 31
5.4.1 Analysis of Sample and Hold circuit 31
5.4.2 Analysis of Comparator 32
5.4.3 SAR ADC Central logic 32
5.4.4 Top level 7 bit SAR ADC Analysis 33
5.5 IMAC 4-bit by 4-bit multiplication Analysis 34
5.6 IMAC 4X4 Array Analysis 35
5.7 Performance Metrics 36
5.7.1 Timing analysis 36
5.7.2 Power consumption 36
5.7.3 Power Delay Product 36

5.7.4 Throughput 36
5.7.5 Energy Efficiency 36
5.8 Results of praposed work 37
5.9 Trade-offs and limitations of proposed methodology 37

6 Conclusion and Future scope **38**
6.1 conclusion 38
6.1.1 Key contributions include: 38
6.1.2 Key findings: 38
6.2 Future scope 39

Index of Terms

ADC: Analog-to-Digital Converter
AI: Artificial Intelligence
BL: Bit Line
WL: Word Line
C-DAC: Capacitor-Based Digital-to-Analog Converter
CMOS: Complementary Metal-Oxide-Semiconductor
DAC: Digital-to-Analog Converter
SRAM: Static Random Access Memory
DRAM: Dynamic Random-Access Memory
SAR ADC: Successive Approximation Register ADC
GOPS: Giga Operations Per Second
TOPS/W: Tera Operations Per Second per Watt
IMC: In-Memory Computing
LSB: Least Significant Bit
MSB: Most Significant Bit
MAC: Multiply-and-Accumulate
VDD: Supply Voltage
VMAC: Voltage representing Multiply-and-Accumulate result
VPS: Virtual Power Supply (summation line)

Chapter 1

Introduction

A **Multiply-Accumulate** (MAC) unit is a core arithmetic building block commonly used in digital signal processing (DSP), machine learning, and other computationally intensive applications. The MAC unit is responsible for performing two fundamental operations:

- Multiplication: It computes the product of two input operands, typically binary numbers.
- Accumulation: The resulting product is added to a previously stored or ongoing sum.

The MAC operation is a cornerstone in many algorithms that require repetitive arithmetic over large datasets, such as convolution operations in neural networks, filtering in DSP, and matrix multiplications in linear algebra. The efficiency of a MAC unit directly influences the performance of these applications, making its design critical in high-speed, low-power systems.

In-Memory Computing (IMC) represents a novel computational paradigm aimed at overcoming the limitations of traditional von Neumann architectures, where data must constantly move between memory and the central processing unit (CPU). This frequent data transfer creates what is known as the "memory wall," leading to latency issues and increased energy consumption. IMC eliminates this bottleneck by embedding computation capabilities directly within memory arrays. Instead of transferring data to a separate processor, arithmetic and logic operations are performed inside the memory itself. This approach significantly reduces data movement, enhances processing speed, and improves energy efficiency.

IMC architectures enable a high degree of parallelism and are particularly effective for data-centric applications such as artificial intelligence (AI), machine learning (ML),

and real-time signal processing. By exploiting emerging memory technologies and advanced circuit-level techniques, IMC provides a scalable and energy-efficient alternative to conventional computing systems, aligning well with the growing demands of modern workloads.

1.1 Background

Memory is a fundamental component of any computing system, serving as the primary storage location for data and instructions that the processor needs to execute tasks. Memory architecture plays a crucial role in determining the performance and efficiency of computing systems. Traditional architectures, such as the von Neumann and Harvard models, define how data and instructions are stored and accessed, each with distinct advantages and limitations.

Von Neumann Architecture The classical and most widely used architecture where program instructions and data share the same memory space. A single bus is used for both data and instructions, and the CPU fetches instructions and data sequentially.

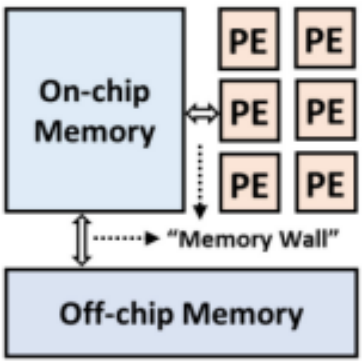

Figure 1.1: Von-Neumann Arch.
Ref:A Brain-Inspired ADC-Free SRAM-Based In-Memory Computing Macro With High-Precision MAC for AI Application(DOI: 10.1109/TCSII.2022.3224049)

- Characteristics:
  Unified memory for data and instructions
  Sequential access can cause bottlenecks (memory wall).
  Simple and flexible design.

- Limitations:
  The shared bus can become a bottleneck.
  Memory access delays limit processor performance

Memory Wall Bottleneck in Von Neumann Architecture

The memory wall refers to the significant delay and energy consumption caused by the continuous and frequent data transfer between the processor and memory. As processor speeds have increased exponentially over the years, memory access speeds have not kept pace, resulting in a growing disparity. Consequently, the CPU often remains idle, waiting for data to be fetched or written back, which limits overall system performance. This bottleneck not only causes latency but also leads to increased power consumption due to the large volume of data movement, making the architecture inefficient for modern data-intensive applications such as artificial intelligence, machine learning, and big data analytics.

How In-Memory Computing (IMC) Overcomes the Memory Wall?

In-Memory Computing refers to a class of architectures where certain computational tasks are performed directly within the memory arrays, leveraging memory elements not just for storage but also for processing. By integrating compute capabilities into memory circuits, IMC reduces the need for data movement, lowers power consumption, and enhances throughput. This approach is particularly well-suited for operations that are inherently parallel and repetitive, such as bitwise logic, addition, multiplication, and accumulation.

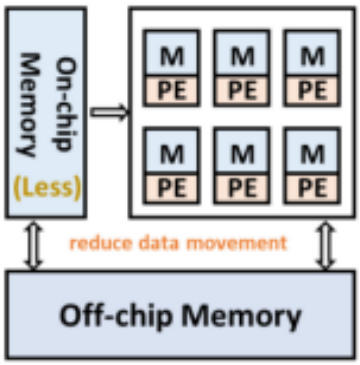

Figure 1.2: In-memory computing Arch.
Ref:A Brain-Inspired ADC-Free SRAM-Based In-Memory Computing Macro With High-Precision MAC for AI Application(DOI: 10.1109/TCSII.2022.3224049)

1.2 Motivation

The motivation behind this work is to design and explore an In-Memory MAC unit that can effectively combine storage and computation into a unified architecture. Such a design not only addresses the inefficiencies of traditional systems but also provides a scalable and energy-efficient solution for next-generation computing platforms.this research seeks to provide an innovative and improved solution that surpasses the limitations of existing approaches.

The design aims to demonstrate improved performance in terms of speed, energy efficiency, and computational density, particularly for data-intensive applications such as artificial intelligence, machine learning, and digital signal processing.

This work is driven by the need to minimize the data transfer bottleneck between memory and processing units that is inherent in traditional systems. By performing MAC operations directly within the memory array—especially using SRAM-based IMC architecture—the goal is to significantly reduce latency and power consumption while enabling parallel computation.

1.3 Problem Statement

Current-domain SRAM-based IMC faces challenges like non-linearity due to process variation and high dynamic power consumption. There is a need for a more linear and energy-efficient solution that supports charge-sharing MAC operations directly within memory

1.4 Objective of the Work

- To analyze and compare traditional MAC unit architectures and their associated inefficiencies in modern workloads.
- To explore SRAM-based IMC architectures suitable for implementing arithmetic operations within the memory structure.
- To design a MAC unit that performs multiply and accumulate operations directly in the memory using bit-line or word-line based computation methods.
- To evaluate the performance of the proposed IMC-based MAC unit in terms of area, power, delay, and scalability.
- To demonstrate the advantages of the IMC-based MAC unit over traditional logic-based implementations for data-intensive computing tasks.

1.5 Organisation of the Thesis

- **Chapter 1: Introduction**
  Introduces the background and motivation for the research. It defines the problem statement, objectives, and the scope of the thesis.

- **Chapter 2: Literature Review**
  Reviews previous work in the area of In-Memory Computing. It compares different memory technologies such as SRAM, DRAM, RRAM, PCM, and highlights the research gap targeted by this work.

- **Chapter 3: Proposed Methodology**
  Describes the overall system architecture. It includes the design flow, block diagram, technology choice, and operation principle of the core components including the 9T1C SRAM bitcell, C-DAC, and SAR ADC.

- **Chapter 4: Design and Implementation**
  Details the EDA tool flow using Cadence and provides the schematic design, simulation, and integration of all major blocks such as address decoder, bitcell, DAC, ADC, and the 4×4 MAC array.

- **Chapter 5: Results and Discussion**
  Presents the simulation results and waveform analysis of each module. It evaluates performance metrics such as power, delay, throughput, and energy efficiency. Limitations and design trade-offs are also discussed.

- **Chapter 6: Conclusion and Future Scope**
  Summarizes the key contributions and findings. It outlines the limitations of the current implementation and proposes directions for future work including layout-level design, post-layout simulations, and ASIC fabrication.

Chapter 2

Literature Review

2.1 Evolution of IMC and Architectural Shifts

This chapter provides a detailed survey of various In-Memory Computing (IMC) methodologies. It includes approaches based on Static and Dynamic RAM, emerging non-volatile memory (NVM) technologies. Initial developments in IMC were inspired by the need to handle massive data processing requirements in AI and machine learning. Early work focused on processing-in-memory (PIM) architectures, where auxiliary logic was placed near memory. As technology evolved, researchers proposed compute-in-memory (CIM) architectures, which perform operations directly within memory cells using analog or digital techniques**[13]**
Several memory technologies have since been explored to implement IMC:

- SRAM-based IMC: Preferred for high-speed, low-latency applications.**[1]**, **[2]**, **[4]**, **[5]**, **[7]**, **[11]**
- ReRAM and PCM-based IMC: Suitable for analog computation and neuromorphic applications**[8]**, **[12]**, **[17]**
- DRAM-based IMC: Offers high density but is volatile and energy-intensive**[13]**

2.1.1 Digital In-Memory Computing

In this approach, Boolean operations (AND, OR, XOR) and arithmetic functions are performed using standard digital logic inside or near the memory arrays. SRAM is commonly used here due to its CMOS compatibility and fast access. Digital IMC leverages standard SRAM or DRAM bitcells, enhanced with peripheral logic, to enable efficient Boolean and arithmetic operations within the memory itself. Unlike analog IMC approaches, digital IMC offers improved scalability, noise immunity, and compatibility with existing CMOS technology and design flows**[1]**, **[11]**. It significantly improvements in energy efficiency, throughput.

work flow of Digital IMC

1. Data Storage:Input operands (like weights and activations) are stored in the SRAM array**[1]**, **[7]**.
2. Bitline Activation: Wordlines are selectively enabled to access multiple memory cells in parallel.
3. Bitwise Multiplication:For binary MAC, multiplication is usually implemented using AND gates or controlled discharge of bitlines.Partial products are generated in parallel.
4. Accumulation:Accumulated either in peripheral logic (digital adders) or in-memory (using sequential logic or counters).Adders may be placed near the memory bank to minimize data movement.
5. Output:Final accumulated result is stored back in memory or sent to the next stage.

Benefits of Digital IMC

- No need for ADC/DAC (unlike analog IMC).
- More robust to noise.
- Easier integration with standard digital design flows.
- Bit-precision control is straight forward **[11]**, **[13]**.

2.1.2 Analog In-Memory Computing

Analog In-Memory Computing performs mathematical operations (like multiply and accumulate, MAC) directly in the analog domain, within memory arrays, using electrical signals such as voltage or current. This eliminates data movement between memory and processing units, leading to energy efficiency**[2]**, **[3]**, **[5]**, **[8]**.

Workflow of analog IMC:

1. Store Weights as Conductance.
2. Apply Input as Voltage.
3. Compute Current via Ohm's Law
4. Sum Currents via Kirchhoff's Law
5. Output Result

2.2 Comparative Analysis of Digital IMC & Analog IMC

Feature	**Analog IMC**	**Digital IMC**
Power Efficiency	Very high	High
Precision	Low-Medium	High
Speed	Extremely high	High
Noise sensitivity	High	low
Area overhead	low	higher
ADC/DAC Requirement	yes	no
Scalability	Limited	Good

Table 2.1: Comparison of Analog IMC and Digital IMC Features

2.3 Existing IMC architectures

2.3.1 SRAM-Based In-Memory Computing

Static Random Access Memory (SRAM) is widely adopted in digital systems owing to its fast operation, low latency, and compatibility with standard CMOS processes. In IMC architectures, SRAM can support both digital and analog computing within the memory array, making it suitable for tasks requiring high throughput and minimal delay**[1]**, **[4]**, **[7]**, **[16]**.

To enable efficient in-memory operations, researchers have introduced custom SRAM cell configurations, such as 8-transistor (8T) and 9-transistor (9T) designs. These allow for parallel, non-destructive reads — a fundamental requirement in many IMC systems. An example is the X-SRAM framework, which employs charge accumulation on bitlines to carry out analog multiply-and-accumulate (MAC) operations within modified 8T SRAM arrays. Similarly, the Neural Cache approach leverages standard SRAM caches in processors to perform digital MAC operations using a bit-serial scheme for AI inference workloads.

Although SRAM-based IMC offers seamless integration with existing digital logic and benefits from CMOS compatibility, it faces limitations such as restricted analog precision, increased area due to peripheral circuitry like DACs and ADCs, and vulnerability to process variations. Nonetheless, due to its maturity and ease of implementation, it remains a widely used platform for practical IMC solutions**[2]**, **[4]**, **[5]**

2.3.2 DRAM-Based In-Memory Computing

Dynamic Random Access Memory (DRAM) is the predominant choice for main memory in computing systems. Recent research has shown that conventional DRAM chips can be repurposed for logic-in-memory operations with minimal hardware modifications. DRAM-based IMC exploits the capacitive nature of DRAM cells and enables logic computations through specialized row-level activations.

A prominent example is the Ambit architecture, which achieves in-memory logic by activating three DRAM rows simultaneously to perform bitwise operations like AND and OR. Another technique, RowClone, accelerates data copy and initialization by utilizing DRAM's internal structures for fast, direct memory operations**[7]**.

While DRAM-based IMC benefits from high memory density and low cost, it is typically restricted to digital logic functions and is not well-suited for analog processing. It also requires careful coordination with the memory controller and adherence to DRAM timing constraints, which can complicate implementation

2.3.3 RRAM-Based In-Memory Computing

Resistive RAM (RRAM) is a type of non-volatile memory that stores data using varying resistance levels. Its analog characteristics make it highly conducive to IMC, especially for implementing vector-matrix multiplications (VMM) through Ohm's and Kirchhoff's laws in crossbar array structures.

The ISAAC architecture is a notable example where RRAM-based IMC is used for analog MAC operations, particularly in deep learning inference tasks. Other studies have demonstrated methods for achieving multi-bit precision using digital-to-analog and analog-to-digital converters to interface with the analog computation layer.

Despite offering significant advantages in terms of energy efficiency and computational density, RRAM technologies face challenges such as endurance limitations, non-linearity in conductance changes, and variability across memory cells. These issues must be addressed before RRAM-based IMC can be deployed at scale reliably.

2.3.4 PCM-Based In-Memory Computing

Phase Change Memory (PCM) leverages the distinct electrical resistance between the amorphous and crystalline states of chalcogenide materials. PCM can store multiple bits per cell and supports analog computation, making it a promising candidate for IMC applications.

In PCM-based IMC, analog operations such as VMM are performed through precise tuning of cell conductance using iterative write-verify techniques. One advanced method involves mixed-precision training, where analog computation is used for matrix

operations, while critical updates and adjustments are carried out digitally to preserve accuracy**[12]**, **[13]**.

While PCM offers benefits like high density and multi-level storage, it also presents drawbacks including slow write times, thermal interference, and long-term resistance drift — all of which affect its viability for analog-intensive or frequently updated workloads.

2.3.5 MRAM-Based In-Memory Computing

Magnetoresistive RAM (MRAM), especially the Spin-Transfer Torque (STT) variant, is a non-volatile memory that stores data using magnetic tunnel junctions (MTJs). MRAM's robustness and fast switching capabilities make it suitable for IMC, particularly for digital logic applications**[13]**.

In MRAM-based IMC, MTJs can be configured to function as basic logic gates, enabling bitwise operations such as AND, OR, and XOR directly within the memory array. These in-memory operations are executed during read/write phases, improving speed and energy efficiency.

However, MRAM's use in analog computation is limited due to its binary switching nature. Additionally, MRAM cells tend to be larger compared to other NVMs, which may hinder scaling. As a result, MRAM-based IMC is mainly suited for digital logic tasks requiring non-volatility and endurance.

2.3.6 Flash-Based In-Memory Computing

NAND Flash memory, commonly used for mass storage, has also been explored for IMC, especially for inference workloads where endurance and latency are not critical bottlenecks. Flash-based IMC typically performs analog computations by leveraging multiple threshold voltage states of memory cells.

These systems can carry out dot-product operations using the stored charge levels in memory. However, variability, low write endurance, and relatively slow access speeds limit the practicality of Flash IMC to niche applications where high density and infrequent updates are acceptable.

Recent advancements in 3D NAND technologies have rekindled interest in Flash-based IMC by enabling vertical stacking and tighter integration with logic circuits, particularly for energy-efficient AI inference on edge devices.

2.3.7 Capacitor-Based In-Memory Computing

Capacitor-based IMC utilizes charge storage and redistribution across capacitor arrays to perform computations directly in the analog domain. This approach is particularly

suitable for energy-efficient MAC operations in low-power or edge computing environments.capacitive charge sharing is used to implement analog MAC operations with high energy efficiency. These architectures are well-suited for lightweight tasks such as image or audio signal processing**[3]**, **[5]**, **[6]**.

However, capacitor-based IMC faces constraints including charge leakage, limited data retention time, and sensitivity to environmental noise. Therefore, these systems are best applied in scenarios where computation follows data storage almost immediately.

2.4 Research gap identification

Despite substantial advancements in In-Memory Computing (IMC) over recent years, several fundamental challenges continue to hinder the widespread adoption and practical effectiveness of current architectures. One of the primary concerns lies in the limited analog precision achievable in charge-domain or current-domain multiply-and-accumulate (MAC) operations. **[2]**, **[5]**, **[6]** Many existing approaches, particularly those based on capacitor arrays and conventional SRAM, struggle to maintain signal integrity. Factors such as thermal noise, device mismatch, and quantization artifacts typically constrain the analog precision to around 4 to 6 bits. Unfortunately, this resolution falls short of the 7 to 8 bits commonly required for modern AI inference workloads.

Another major bottleneck in mixed-signal IMC systems is the overhead introduced by digital-to-analog converters (DACs) and analog-to-digital converters (ADCs). High-resolution converters, while necessary for improved accuracy, significantly contribute to increased area, power consumption, and latency. Designs employing full-scale flash ADCs often suffer from poor scalability and energy inefficiency as the bit resolution increases. This compromises the energy savings that IMC architectures are intended to deliver.

In capacitor-based charge-domain computing, charge leakage is also a notable issue. The analog data stored on capacitors tends to degrade over time, particularly in advanced CMOS nodes where leakage currents are more pronounced. This results in reduced reliability and accuracy, especially when computations span multiple cycles or require longer accumulation durations.

Furthermore, many traditional SRAM-based IMC architectures, such as those using 6T or 8T cells, are not inherently designed to support simultaneous memory access and analog computation. Doing so either affects cell stability or demands additional peripheral circuitry, which increases complexity and area.

These persistent limitations form the foundation of the motivation behind the architecture proposed in this thesis. The design introduces a 9T1C SRAM-based IMC framework that decouples memory read operations from analog computation by integrating a compute-dedicated capacitor into the cell. In addition, a binary-weighted, low-cost capacitor DAC (C-DAC) is used for input signal generation, followed by passive charge

summation and SAR ADC to efficiently digitize the output.

Together, these innovations aim to address the key challenges of limited precision, leakage, and conversion overhead. The resulting architecture offers improved linearity, scalability, and energy efficiency, Where performance / watt is a critical metric.

Chapter 3

Proposed Methodology

In this chapter, the proposed SRAM-based In-Memory Computing (IMC) architecture utilizes a charge-domain approach to enable mixed-signal processing that combines both digital and analog techniques. It performs inner product computations by interacting a binary weight vector with an input signal vector. The digital inputs are first converted into analog voltages, which are then accumulated proportionally based on the corresponding binary weight values stored in the memory row. This continuous accumulation process effectively carries out the inner product operation. The key advantages of this method include low power consumption, high computational speed, and a reduced need for complex logic circuits, making it particularly efficient for edge AI applications.

3.1 Block diagram

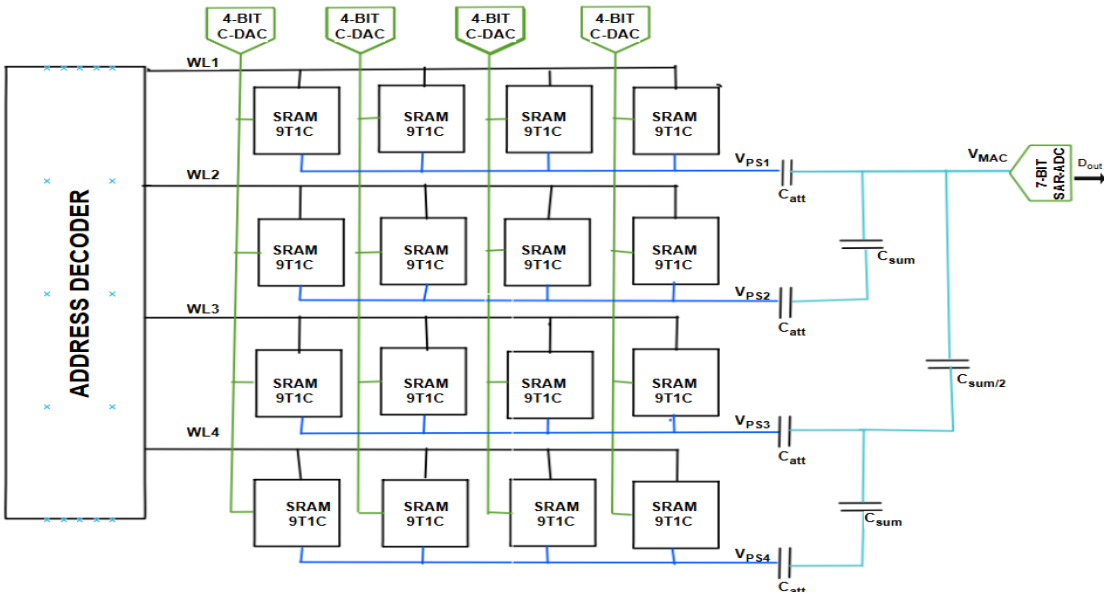

Figure 3.1: Block diagram of proposed architecture

3.2 Design Specifications

Parameter	**Specification**
SRAM Array size	4X4
Input Resolution	4 Bits (Digital)
Output Precision	7 bits (digital)
DAC type	Binary-weighted C-DAC
ADC type	SAR ADC
Technology Node	180nm CMOS
Power Consumption	<400uW

Table 3.1: System Specifications

3.3 Choice of technology

The proposed In-Memory Computing (IMC) architecture has been designed and simulated using the 180nm CMOS technology node, which is well-suited for mixed-signal VLSI applications. This mature and widely-used technology node offers several advantages for analog precision, noise tolerance, and design flexibility, making it ideal for implementing custom SRAM cells, capacitor-based DACs, and ADCs.

3.4 Operation Principle

The proposed In-Memory Computing (IMC) system operates on the principle of charge-domain vector-matrix multiplication (VMM), where digital input activations are converted to analog voltages, modulated by stored binary weights inside SRAM cells, summed in the analog domain using capacitors, and finally digitized using a SAR ADC. This methodology avoids the energy and latency penalties associated with frequent data movement between memory and computation units in conventional von Neumann architectures. .

3.4.1 Word line Activation via Address Decoder

The computation begins with a 2-bit address input fed into the address decoder, which enables one of the 4 wordlines (WLs) in the 4x4 9T1C SRAM array. This row selection determines which stored binary weight vector will participate in the current MAC operation. Only one WL is activated at a time during a compute cycle, ensuring localized access and controlled current/charge flow.

3.4.2 Input Application through 4-bit C-DACs

Each column of the SRAM array is equipped with a 4-bit binary-weighted Capacitor Digital-to-Analog Converter (C-DAC). These DACs receive 4-bit digital input activations and convert them into analog voltages .The analog voltage corresponding to each input vector element is distributed along a column to all SRAM bitcells within that column. The C-DACs use binary-weighted capacitors to generate voltage levels proportional to the digital inputs. These analog voltages act as activation values in the vector-matrix multiplication.

3.4.3 In-Memory Multiplication in 9T1C SRAM Bitcells

Each 9T1C SRAM cell stores a binary weight (0 or 1) and includes an embedded computation capacitor (Ccell). When the corresponding WL is activated: If the stored weight is '1', the analog voltage VIN from the column DAC is allowed to charge the capacitor inside the bitcell. If the stored weight is '0', the transmission gate is not enabled, and no charge transfer occurs. This mechanism realizes multiplication at the bitcell level:

3.4.4 Charge-Based Summation via Capacitor Network

The analog outputs from all bitcells in the selected row are passively accumulated along the row using a capacitor summation network. Each column connects to two summing capacitors to support full and half-bit weighting. The capacitive network forms a charge redistribution tree that collects and sums all voltage contributions from the activated row. This results in an analog voltage at the end of the row, denoted as VMAC, which represents the accumulated result of the vector-matrix multiplication:

3.4.5 Analog-to-Digital Conversion using SAR-ADC

The final analog voltage VMAC is converted to a digital value using a 7-bit SAR Analog-to-Digital Converter which Uses a successive approximation method with a binary search to determine the lower bits. Digital Output Generation Each row produces a 7-bit digital output representing the final MAC result. With multiple rows processed sequentially or in parallel (if multiple WLs are enabled), a full matrix-vector multiplication can be realized efficiently within the memory array.

3.5 Component Level Explanation

The system is comprised of five main functional blocks: an address decoder, binary-weighted capacitor DACs, 9T1C bitcells, a charge-based summation network, and SAR analog-to-digital converter (ADC).

1. **Address Decoder:**This module selects one of 4 wordlines based on a 2-bit binary address input, enabling specific rows of the SRAM array for participation in computation. It provides the selection logic needed for targeted read and compute operations during the VMM process.

2. **Binary-Weighted Capacitor DACs (C-DACs):** Four 4-bit DACs are used to convert digital inputs into analog voltages. These DACs use binary-weighted capacitors with values proportional to 8C,4C,2C,1C corresponding to each bit of the 4-bit input word. The DAC output voltages serve as analog activation inputs and are routed to columns in the 9T1C array.

3. **9T1C SRAM Bitcells:** The custom-designed 9T1C bitcell consists of a 6T SRAM core for binary weight storage, a transmission gate, and a storage capacitor Ccell. When activated, the transmission gate allows the input voltage VIN to charge or discharge the capacitor based on the stored weight, performing analog multiplication. The output voltage from each bitcell represents the product of the input activation and the stored weight.

4. **Capacitor-Based Summation Network:** The outputs of individual bitcells along a row are accumulated using capacitive voltage division. A hierarchical capacitor tree performs summation for 4-bit weighted rows using an 8:4:2:1 capacitor ratio. The use of capacitive coupling significantly reduces summation area and energy compared to switched-capacitor networks.

5. **7-bit SAR ADC :** The final analog MAC result is digitized by an efficient successive approximation register (SAR) architecture. This approach balances speed and energy, enabling conversion in half the cycles compared to conventional ADCs, while maintaining a resolution of 7 bits.

Chapter 4

Design and Implementation

4.1 Tool flow and EDA Tool used

4.1.1 Schematic Design (Cadence Virtuoso Schematic Editor)

All circuit-level blocks—including the 9T1C SRAM cell, C-DACs, capacitor summation network, and SAR-ADC—were designed at the transistor level.Transistor models, passive components (capacitors, resistors), and supply elements were instantiated using the 180nm PDK.Careful sizing was done for NMOS/PMOS widths and lengths (W/L),Capacitor values for DAC/ADC matching, transmission gates, and bitline drivers.

4.1.2 Symbol Creation and Hierarchical Integration

After verification, each sub-block (e.g., SRAM cell, DAC, ADC) was encapsulated as a symbol for top-level integration.The top-level system schematic connected:DACs to each column input ,Wordline decoder to SRAM array rows ,Capacitor summation network to row outputs ,ADCs to each summation node.This modular and hierarchical design allowed efficient reuse and simulation of the entire system.

4.2 Functional Simulation and Analysis (Cadence - ADEL)

Functional simulations were performed to verify SRAM cell write/read stability ,Correct operation of the C-DAC and capacitor summation, Analog voltage accumulation across bitlines, accuracy of the 7-bit ADC.

Key Analysis Conducted:

- **DC Analysis:** For DAC output levels and biasing.
- **Transient Analysis:** For simulating time-domain behavior of VMM cycles.
- **Parametric Sweeps:** For input combinations and clock variation.

4.3 Schematic Design and Simulations of all circuit level blocks

4.3.1 Address Decoder

In order to perform row-wise operations within the 4x4 SRAM array, an address decoder is required to activate specific wordlines based on the input address. A 2-to-4 line decoder was designed using combinational logic gates that map a 2-bit binary input to one of the 4 output lines. Each output line corresponds to a unique wordline (WL[0] to WL[3]) in the SRAM array. This decoder is implemented using static CMOS NAND logic, ensuring minimal power consumption during idle conditions and robust logic levels under different process and temperature variations.

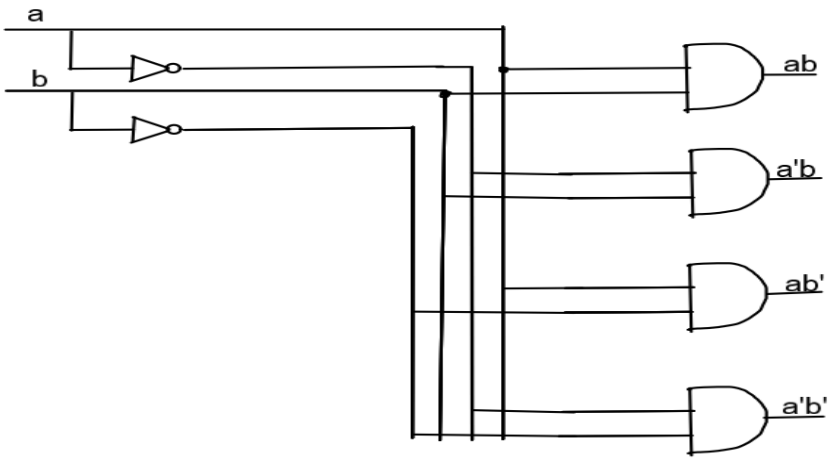

Figure 4.1: Block diagram of Decoder

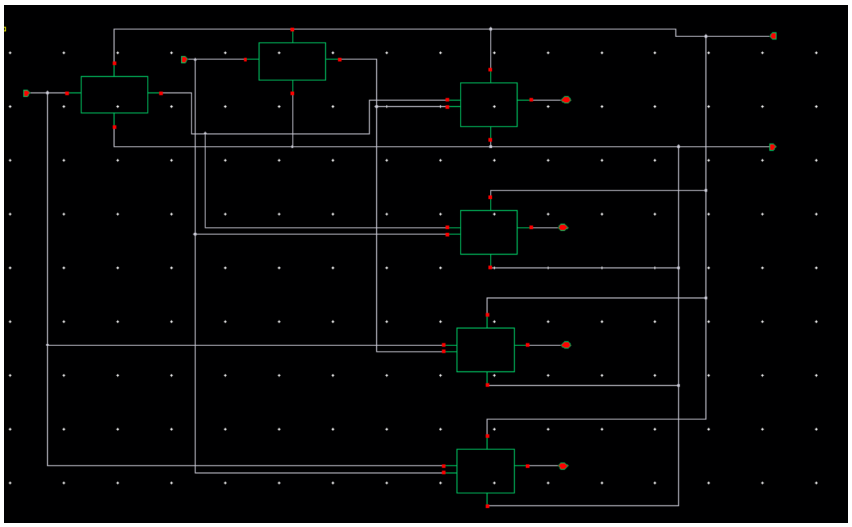

Figure 4.2: Schematic of Decoder

4.3.2 SRAM 9T1C Bitcell

The core of the proposed IMC architecture is the 9-transistor 1-capacitor (9T1C) SRAM cell includes:

- 6T core for standard memory storage.
- 2 additional transistors for separate read/write access to prevent read-disturb issues.
- 1 capacitor connected at the bitline for charge-domain analog MAC operations.

The bitcell accepts analog voltages from the DAC and modulates them based on the stored binary weight, effectively multiplying the two in the charge domain. The layout and connectivity were validated through transient analysis.

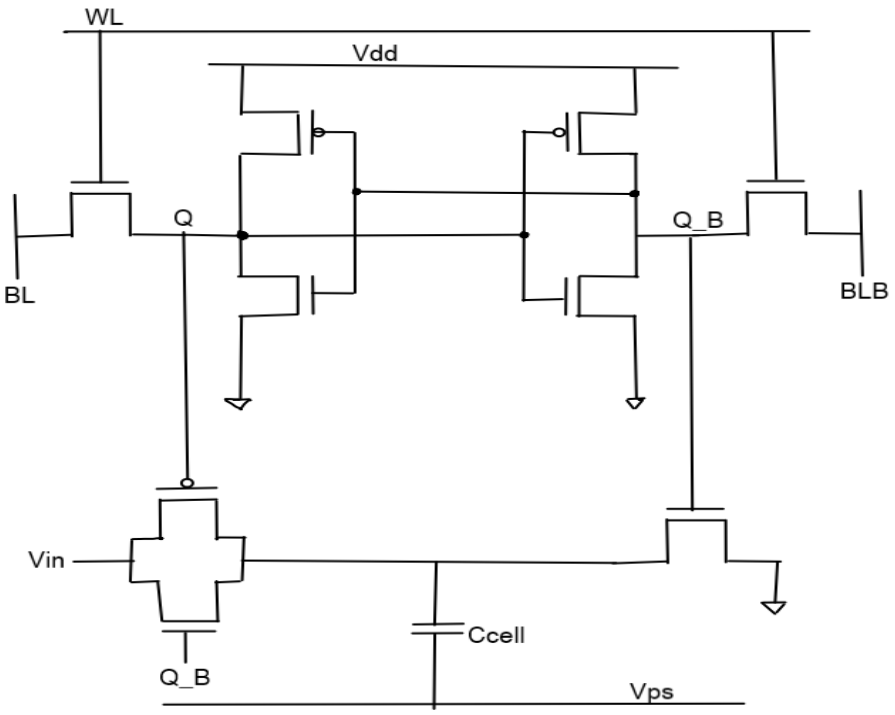

Figure 4.3: Block diagram of SRAM 9T1C

SRAM 9T1C cell consists of

1. 6T SRAM
2. Write Driver
3. Precharge circuit
4. Transmission gate

6T SRAM

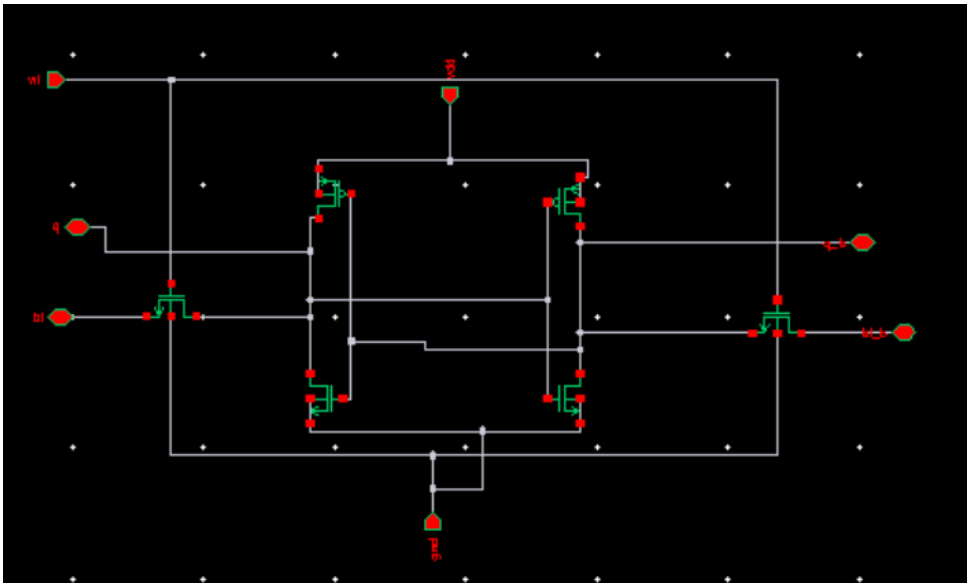

Figure 4.4: Schematic of SRAM 6T

Write Driver

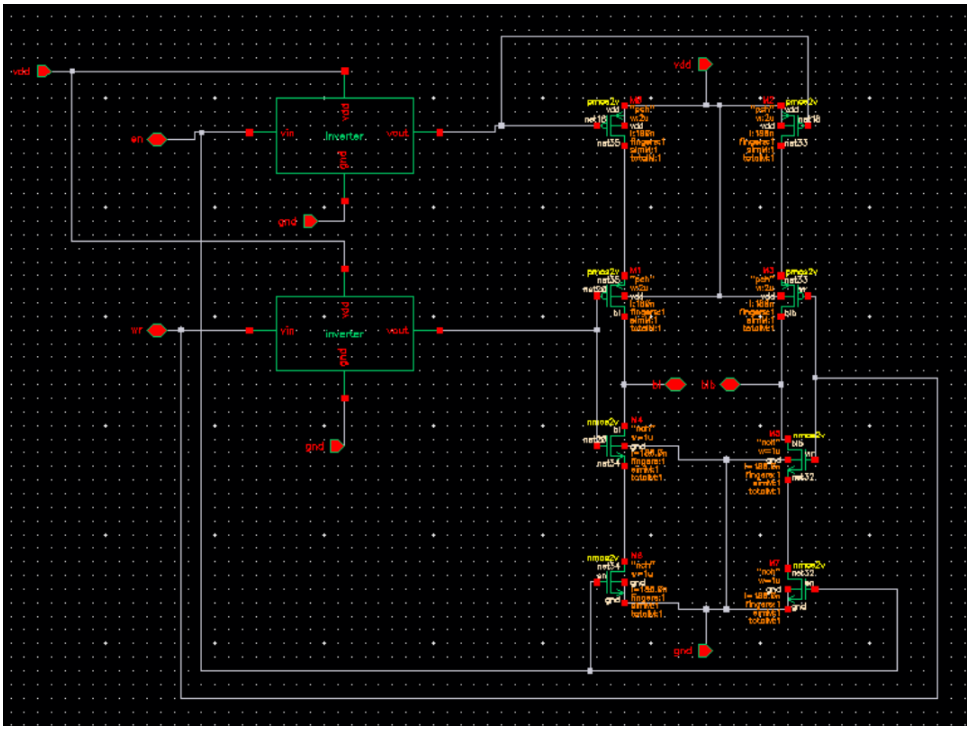

Figure 4.5: Schematic of Write driver

Precharge circuit

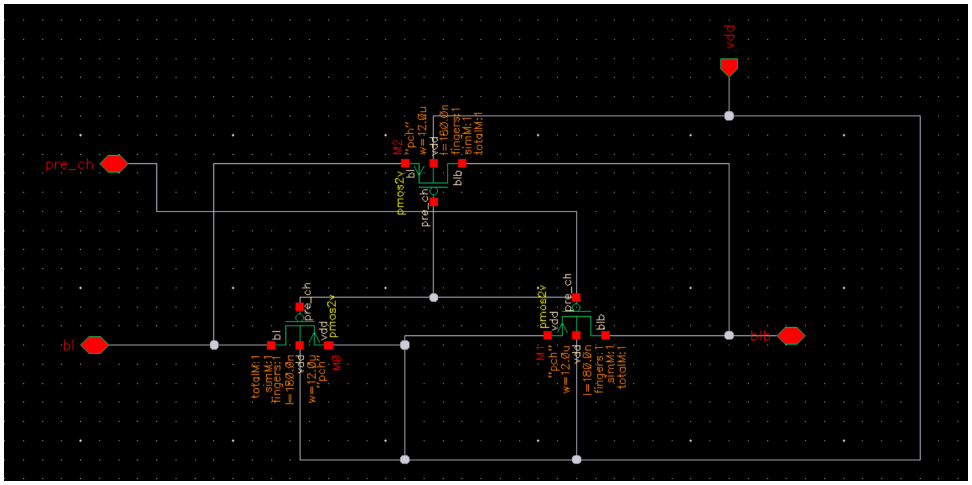

Figure 4.6: Schematic of Precharge circuit

SRAM 9T1C cell

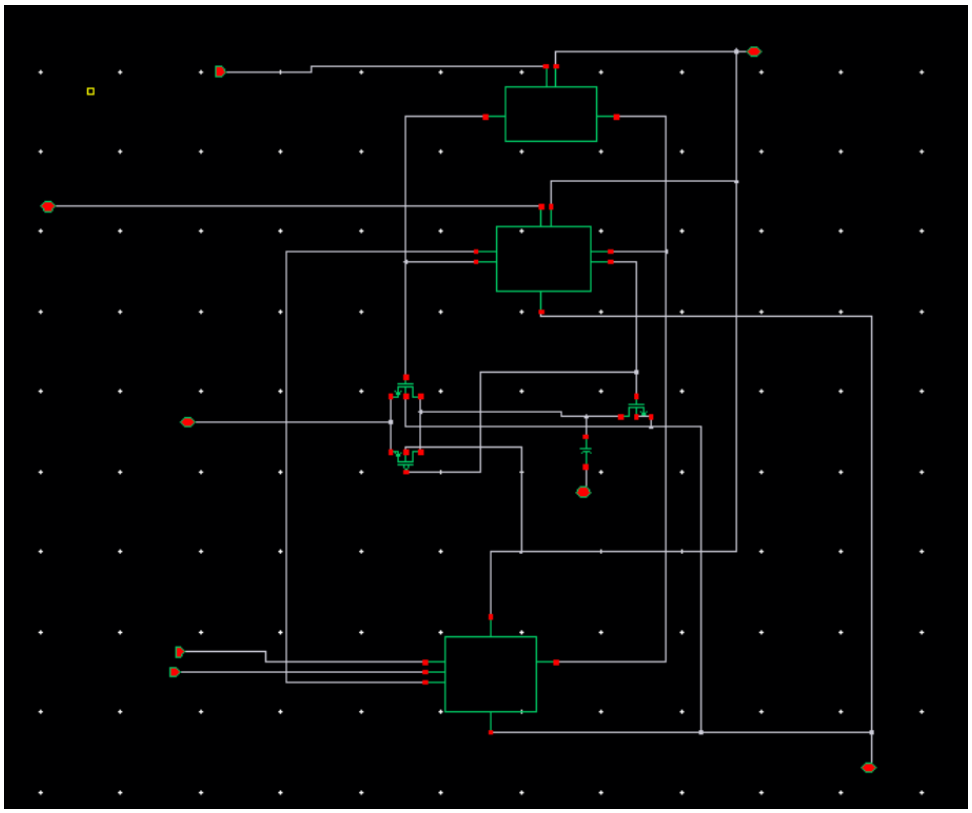

Figure 4.7: Schematic of SRAM 9T1C cell

4.3.3 Binary weighted C-DAC

The DAC takes 4-bit digital inputs and generates corresponding analog voltages ranging from 0 V to 1.8 V. The output of the DAC is directly connected to the gate of access

transistors in the 9T1C bitcells to initiate charge-based multiplication. The capacitor values used are 100fF, 200fF, 400fF, 800fF

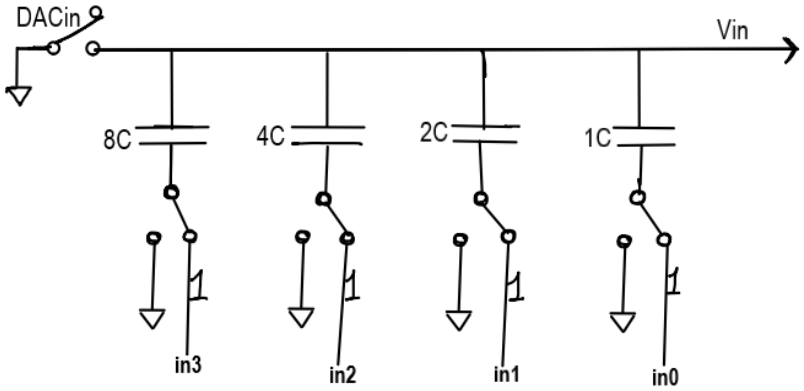

Figure 4.8: Block diagram of praposed C-DAC

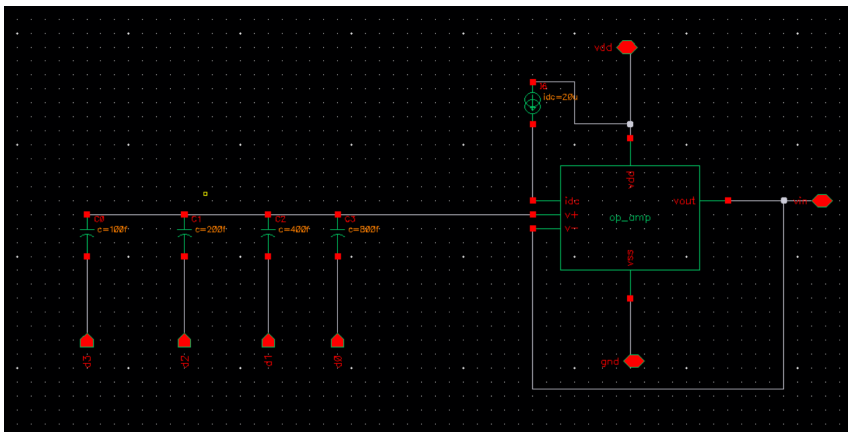

Figure 4.9: Schematic of Binary weighted C-DAC

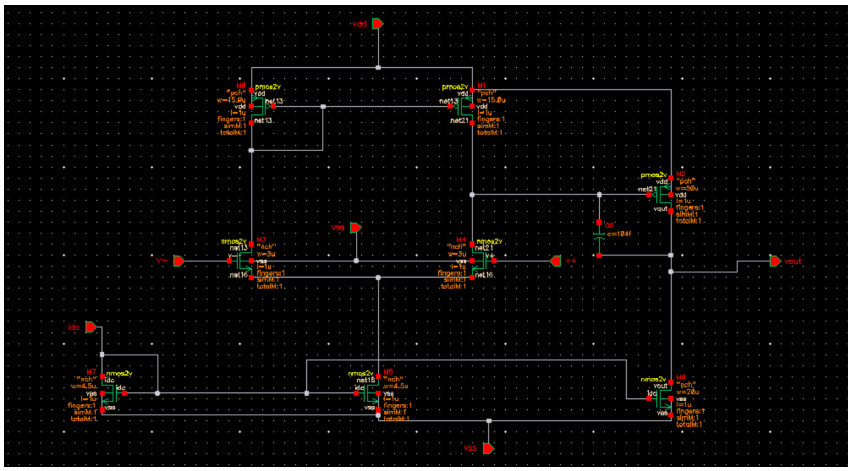

Figure 4.10: Schematic of Operational amplifier

4.4 Successive approximation Register-ADC

The SAR ADC performs conversion by successively narrowing the range in which the input voltage lies. The process begins with the most significant bit (MSB) and proceeds down to the least significant bit (LSB), requiring one clock cycle per bit.

Architectural Components

1. Sample-and-Hold Circuit: Temporarily captures the input voltage.
2. Comparator: Compares the input voltage with the DAC output.
3. SAR Logic: Contains control logic that performs binary search by updating bits.
4. Capacitive DAC (C-DAC): A binary-weighted capacitor array generates reference voltages during conversion.

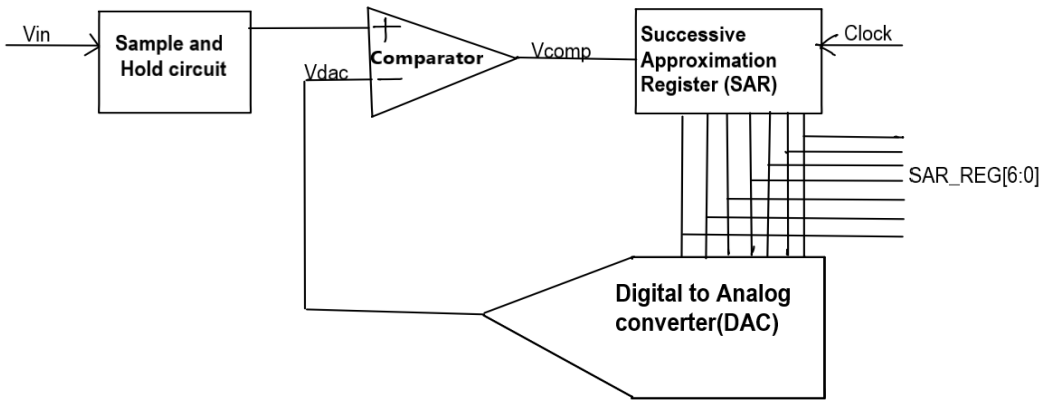

Figure 4.11: Block diagram of SAR-ADC

Sample and Hold circuit

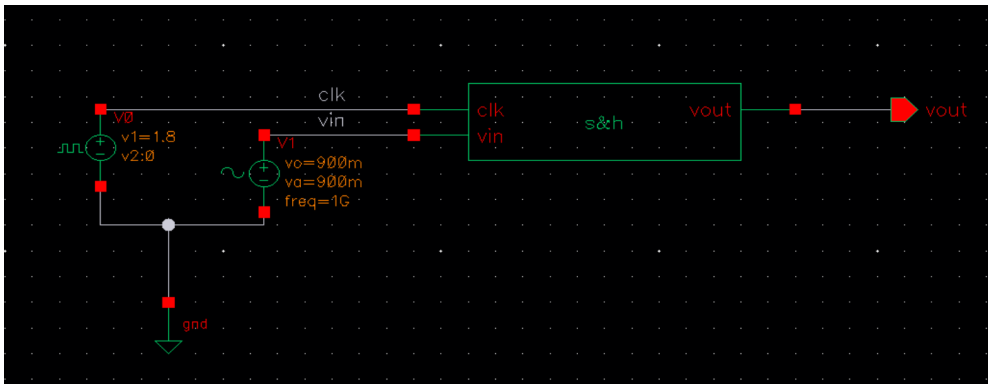

Figure 4.12: Schematic of sample and hold circuit

Comparator

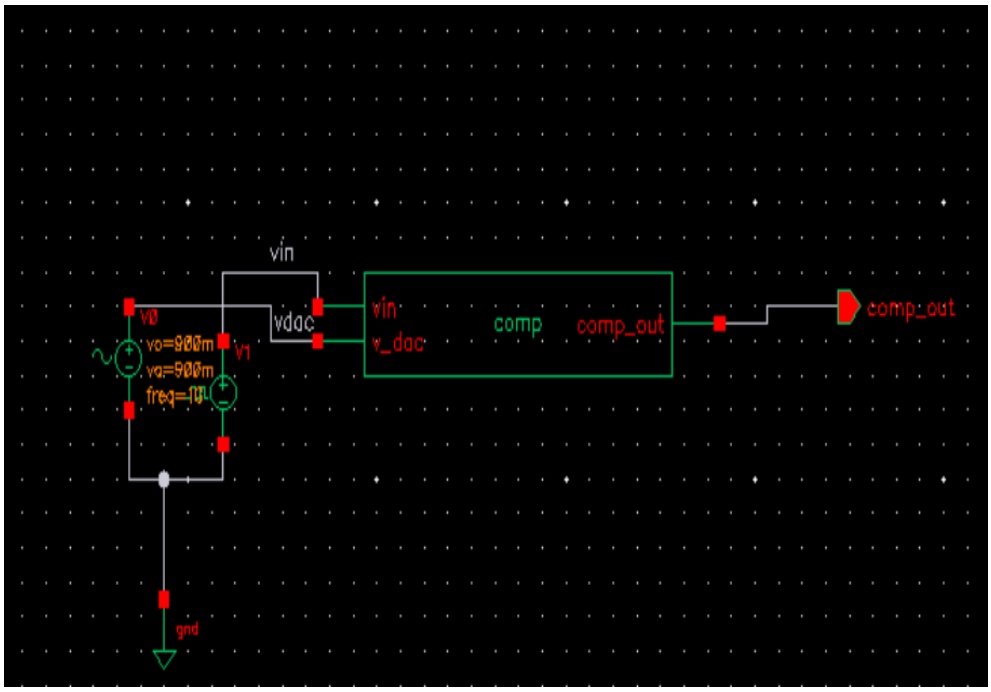

Figure 4.13: Schematic of Comparator

Central logic and SAR register

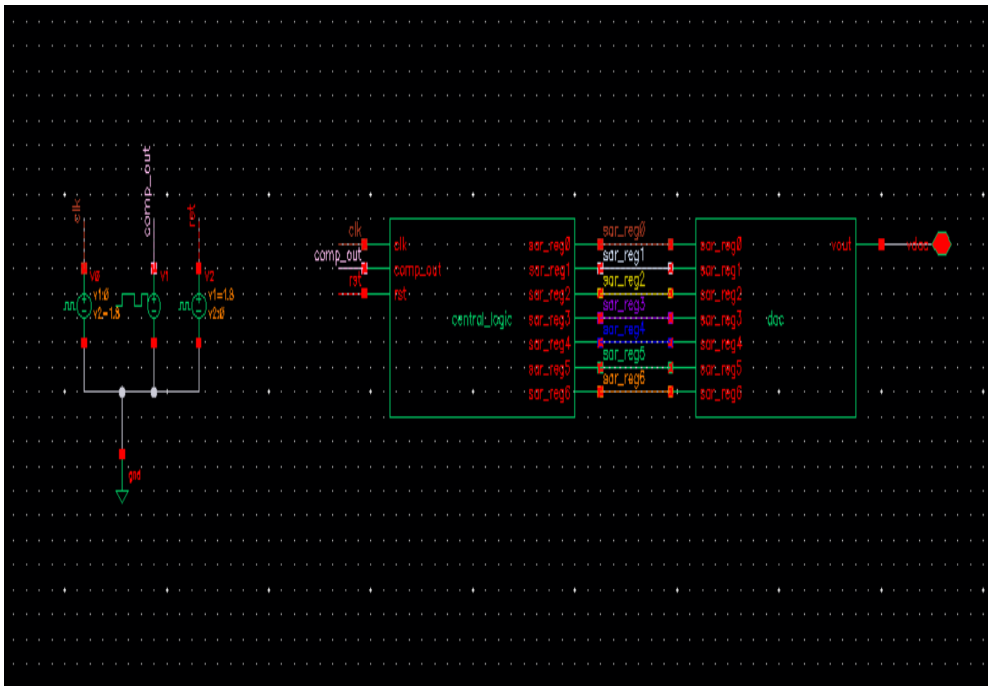

Figure 4.14: Schematic of Integrated Central logic and SAR Register

7 bit SAR-ADC

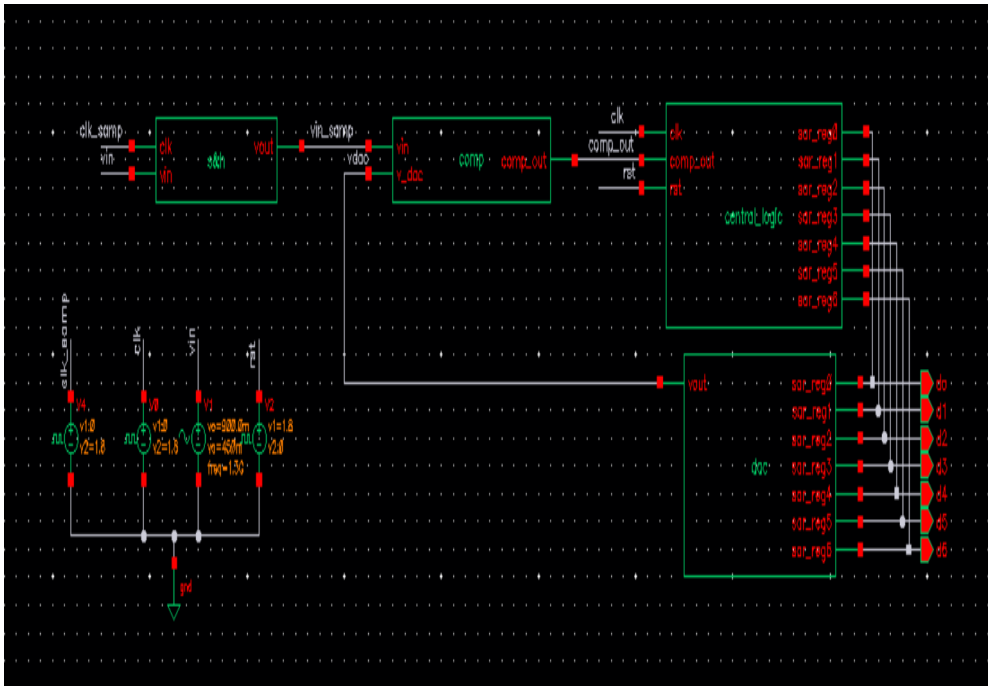

Figure 4.15: Schematic of 7 bit SAR-ADC

4.5 IMAC 4-bit by 4-bit multiplication

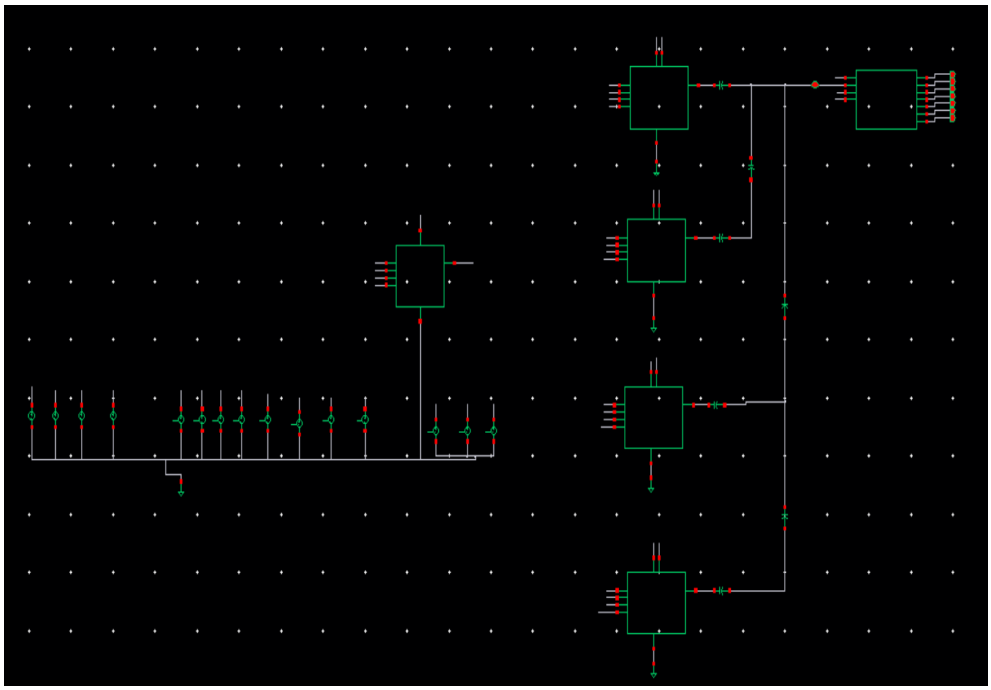

Figure 4.16: Schematic of IMAC 4-bit by 4-bit multiplication

4.6 IMAC 4X4 Array

The schematic shown in the figure below integrates a 4x4 charge-domain SRAM-based in-memory computing system. At the input side, a bank of 4-bit binary-weighted capacitor DACs generates analog voltages based on digital inputs. These voltages are applied to the columns of the 9T1C SRAM array, where each memory cell performs an analog multiplication between the input voltage and the stored bit. A 2-bit address decoder activates one wordline at a time to select a specific row for computation. The resulting charges from the selected row are passed into a capacitor summation array that aggregates the charge using a network of binary-weighted capacitors. The output voltage from the summation is then digitized using a 7-bit SAR ADC. The final digital output represents the result of the multiply-and-accumulate operation for that row.

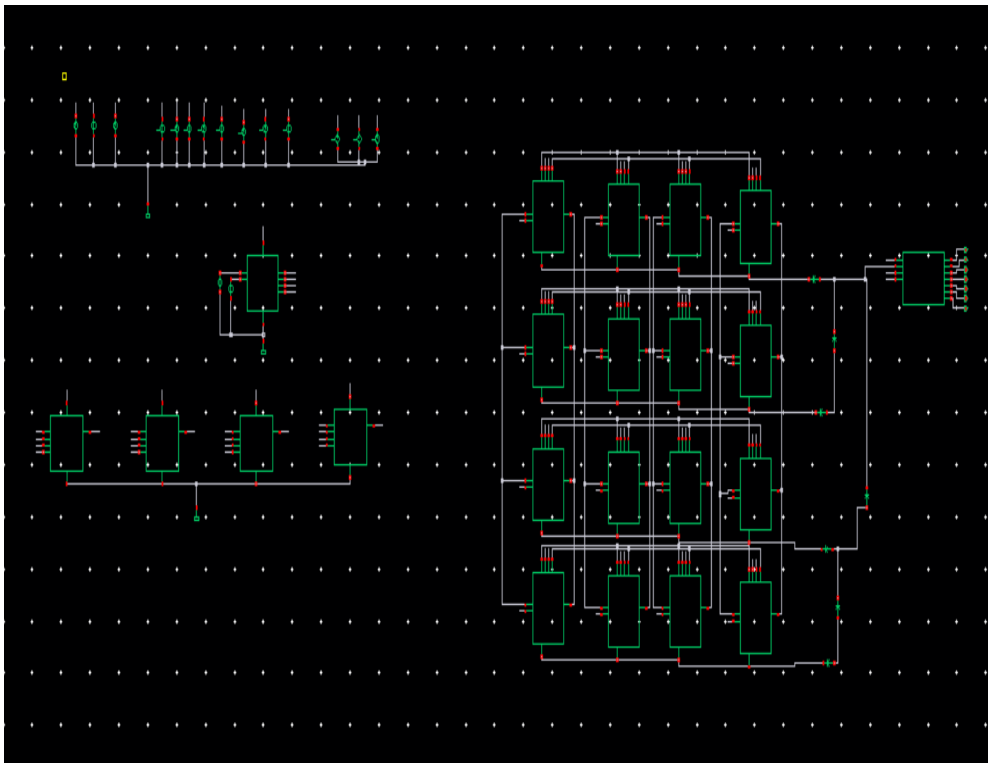

Figure 4.17: Schematic of IMAC 4x4 array

Chapter 5

Results and Discussions

5.1 Decoder Output Waveform and Analysis

The inputs a and b represent a 2-bit binary address, and the outputs wl1 to wl4 correspond to the four decoded lines. As the binary inputs change sequentially from 00 to 11, only one wordline becomes active at each transition, while the rest remain low. This behavior verifies that the decoder performs correct one-hot output generation. Each output line exhibits clean and well-defined transitions from 0 V to 1.8 V, with no overlapping signals or glitches and also shows a consistent timing relationship between input changes and output activation, confirming proper propagation delay through the logic gates.

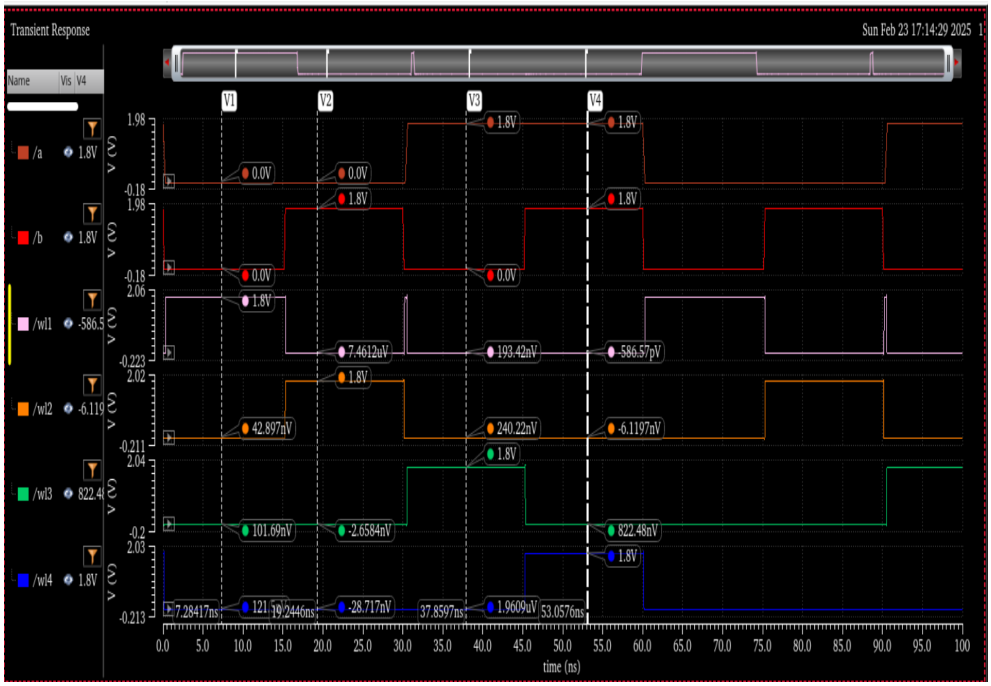

Figure 5.1: Timing Diagram of 2-Bit Address Decoder

5.2 SRAM 9T1C Output Waveforms and Analysis

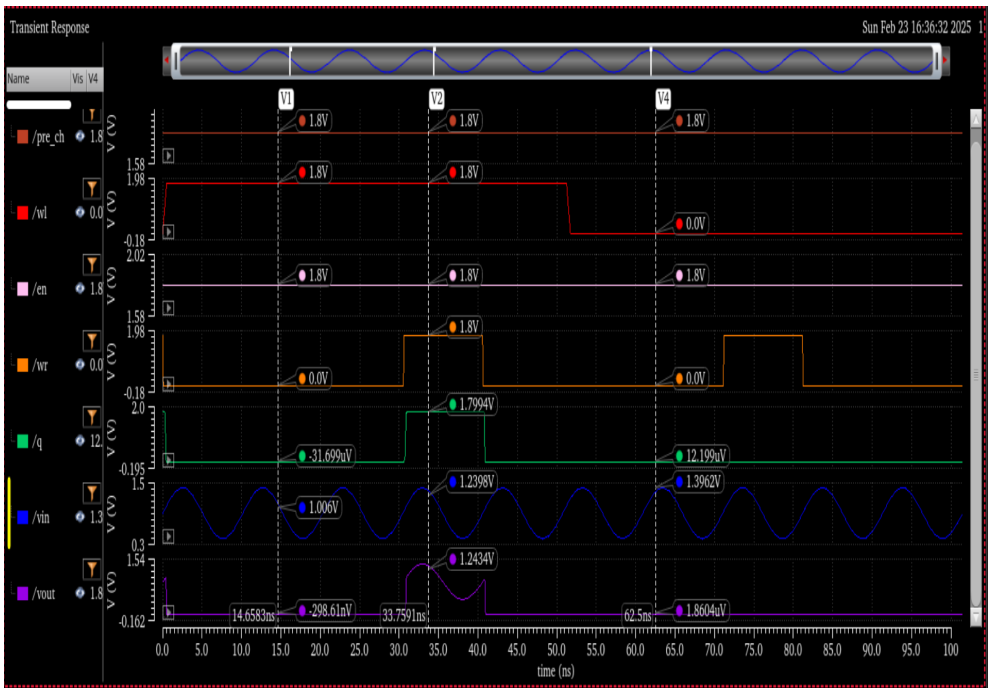

Figure 5.2: Transient Response of 9T 1C SRAM Cell during Read/Write Operation

Above waveform illustrates time-domain behavior of different signals within the SRAM 9T1C cell during a read or write operation. It also illustrates the dynamic characteristics, such as switching times, propagation delays, and the stability of the output. Here reading has started (WL goes high) at 33.75ns and Stabilized value around 35.0 ns.

SRAM read delay = 1.25 ns

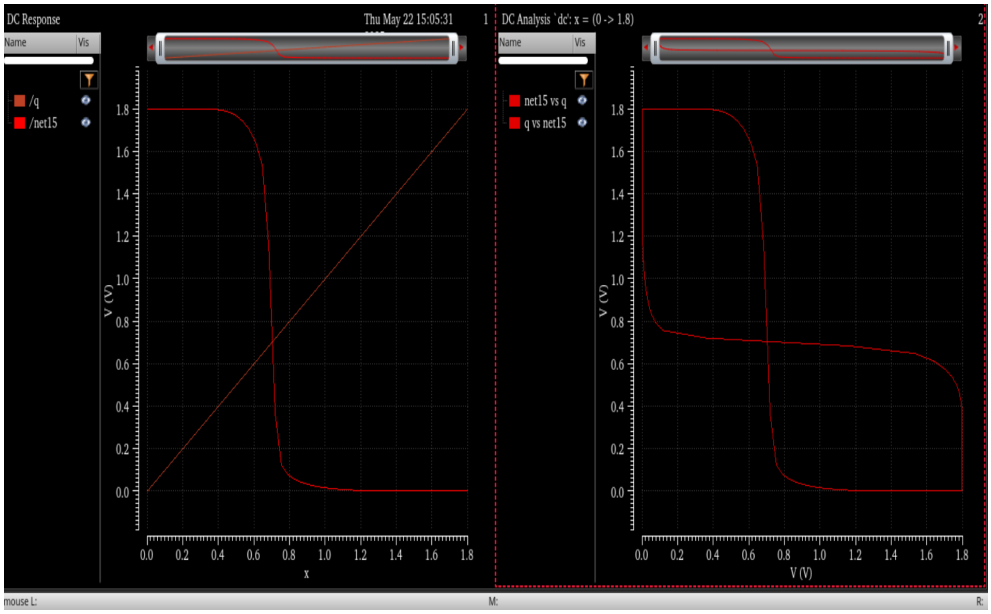

Figure 5.3: Voltage Transfer Curve (VTC) of 9T 1C SRAM Cell

The left plot shows the output voltage (q and net15) as a function of the input voltage (x), illustrating the gain and noise margins of the inverter. The right plot shows the same data with the x-axis representing the output and the y-axis representing the input for analyzing the static noise margin (SNM) and logic levels of the memory cell.

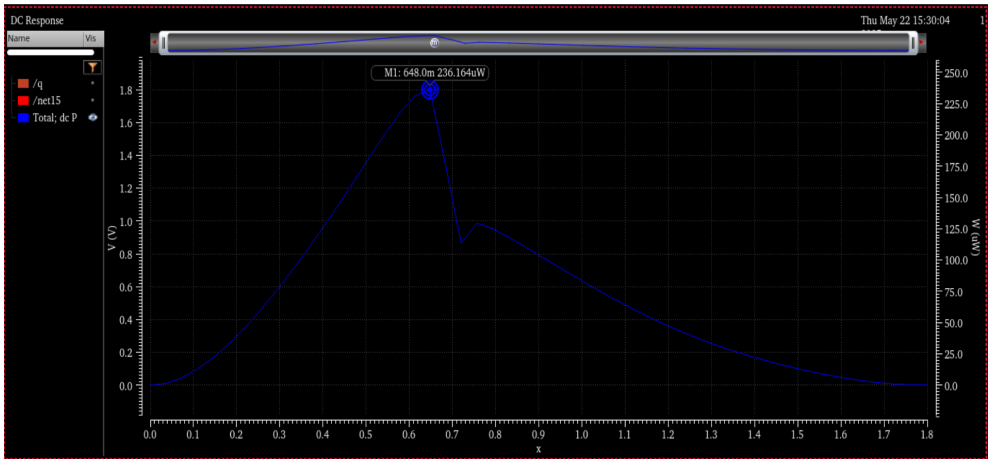

Figure 5.4: Power Dissipation of 9T1C SRAM Cell

The 'M1' marker indicates the peak power dissipation, along with the corresponding input voltage. This analysis is vital for evaluating the power efficiency and consumption of the memory cell.

Power consumption of SRAM cell = 236 µW

5.3 4-bit Binary Weighted Capacitive DAC

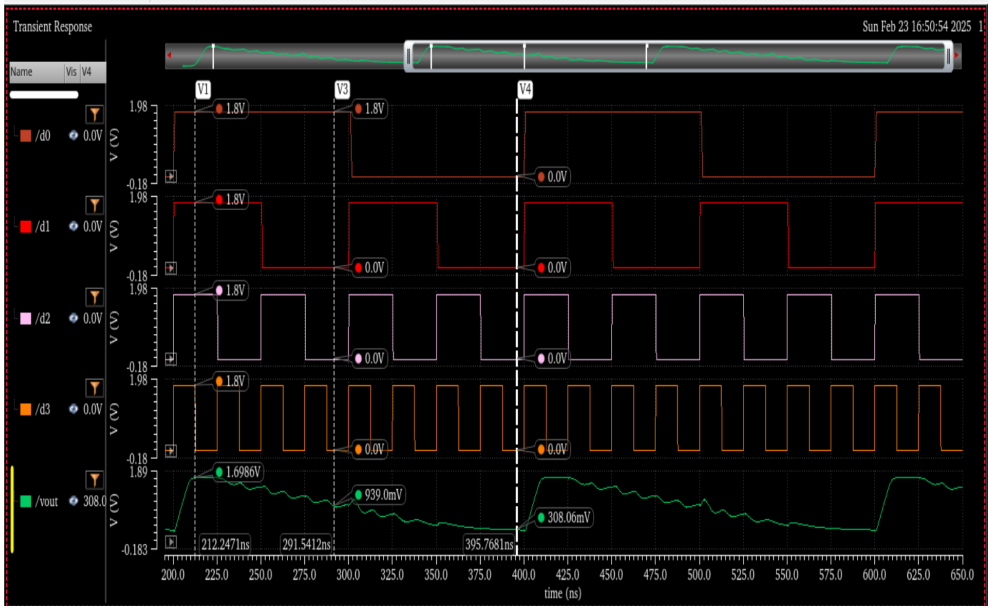

Figure 5.5: Time-Domain Output of Binary Weighted C-DAC

Above simulation shows the expected behavior of a CDAC, converting digital input codes into corresponding analog output voltage levels. Before the transition, Vout is stable at approximately 1.696V.After the transition, Vout settles to approximately 939.0mV.

The first clear transition has occurred at 225ns and output stabilized at 242ns settling time=17ns

5.3.1 Operational amplifier

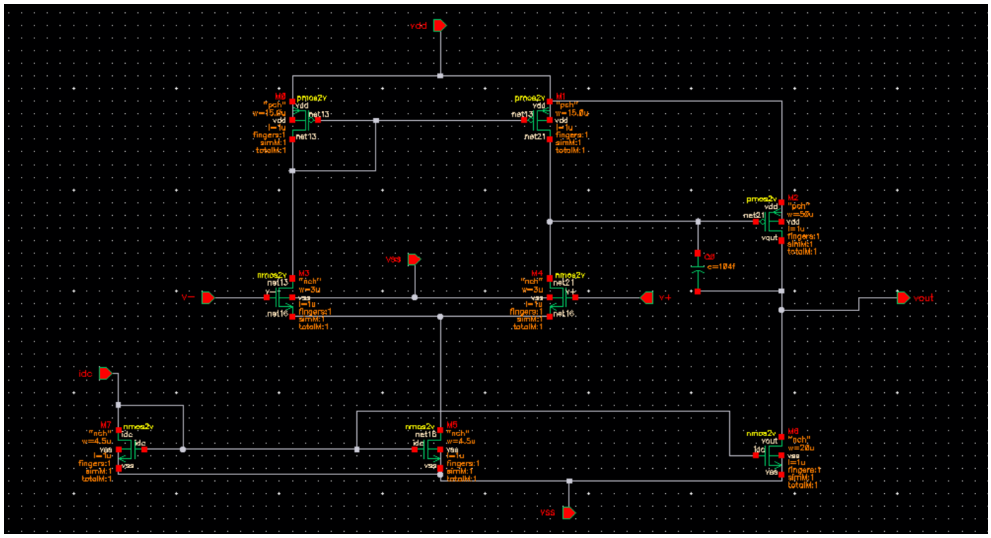

Figure 5.6: Schematic of operational amplifier

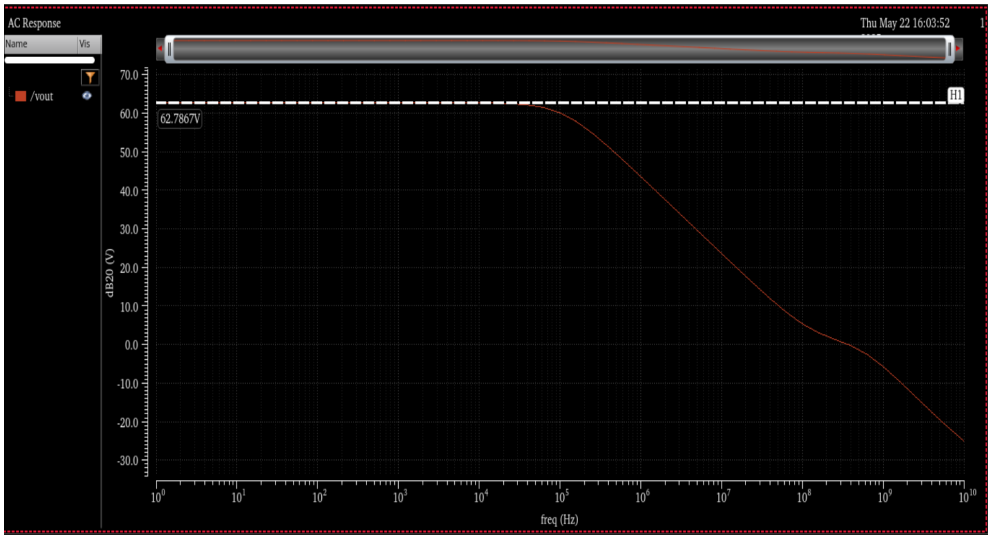

Figure 5.7: Gain response of operational amplifier

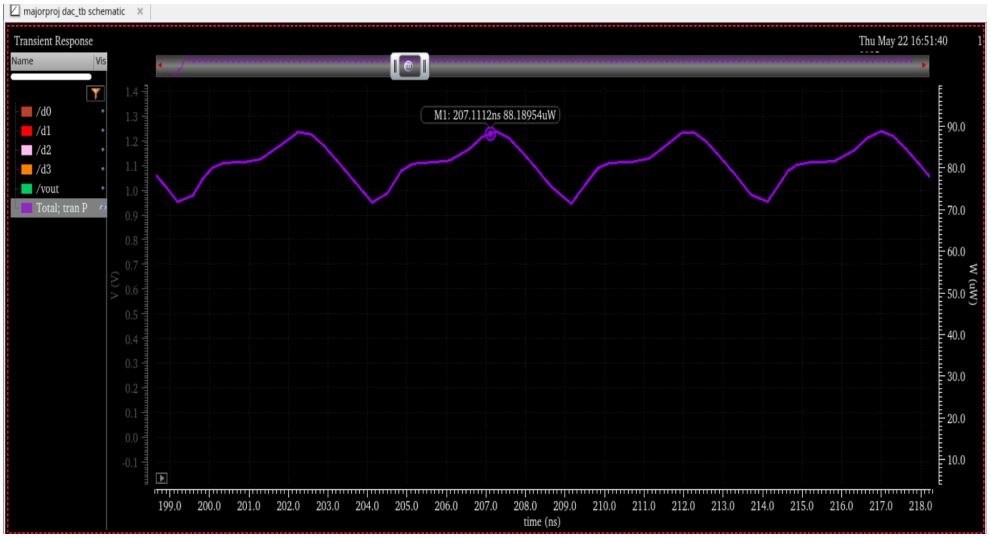

Figure 5.8: Power consumption waveform of C-DAC

The above simulation results demonstrate a functional 4-bit DAC, exhibiting correct digital-to-analog conversion with a fast settling time of around 3-5ns. The op-amp's open-loop gain of approximately 62.78 dB confirms its amplification capability, while the DAC's low power consumption, typically around 90uW highlights excellent power efficiency, consuming only tens to hundreds of microwatts, making it suitable for low-power applications.

5.4 7 bit SAR ADC

5.4.1 Analysis of Sample and Hold circuit

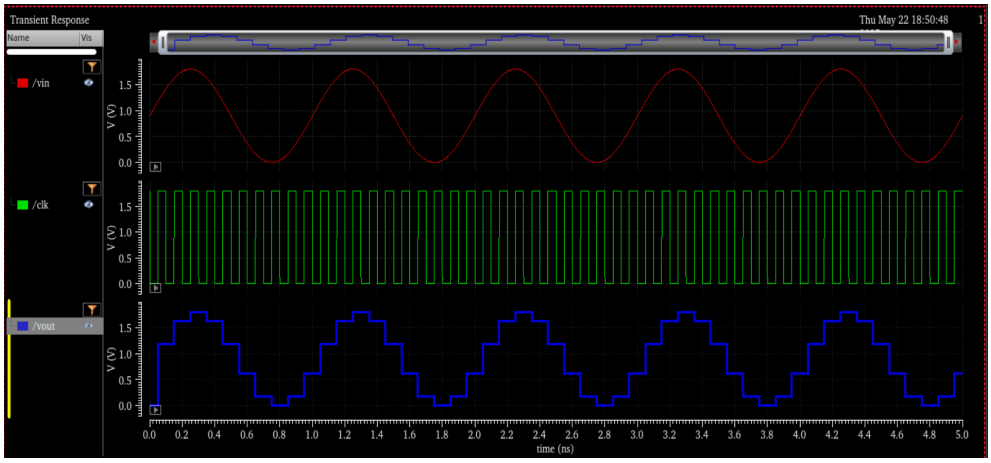

Figure 5.9: Transient Response of Sample-and-Hold Circuit

message signal frequency=1.25GHz
sampling frequency=5GHz

5.4.2 Analysis of Comparator

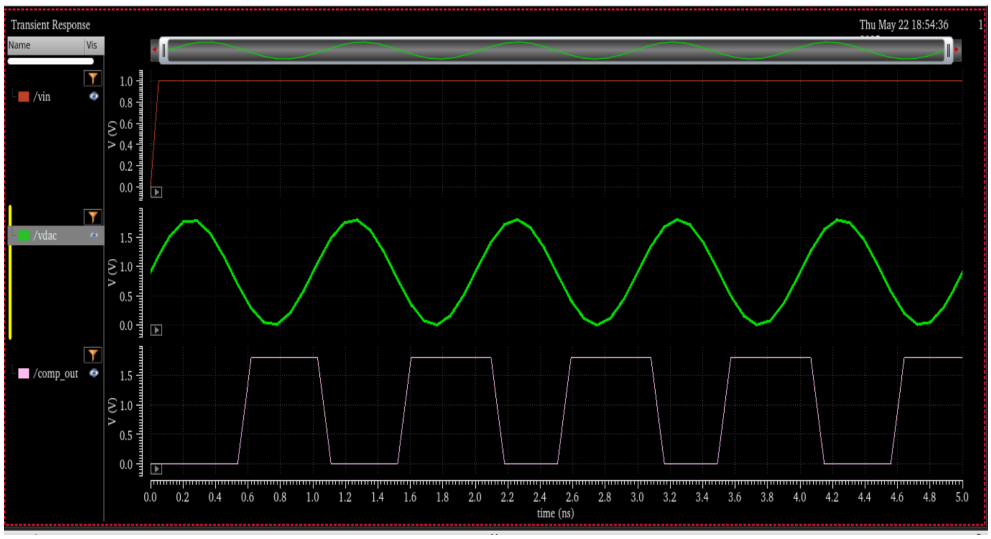

Figure 5.10: output wave form of comparator

5.4.3 SAR ADC Central logic

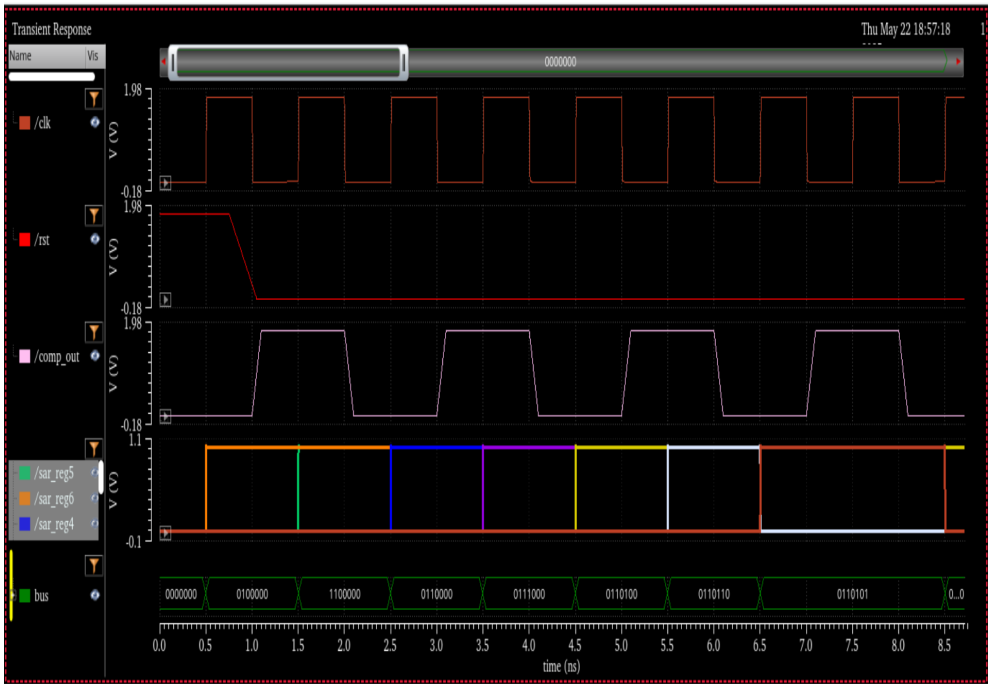

Figure 5.11: output wave form of SAR logic

5.4.4 Top level 7 bit SAR ADC Analysis

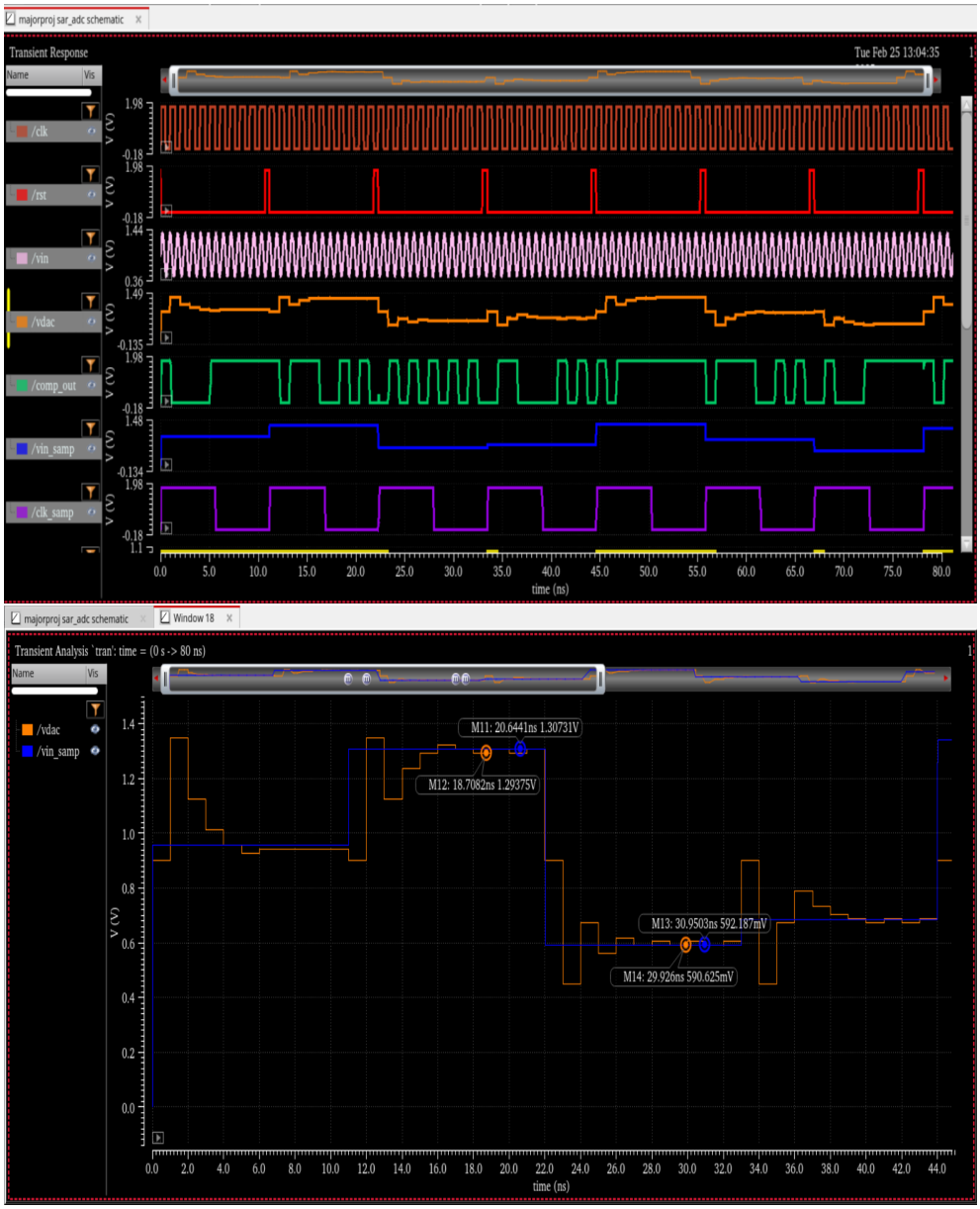

Figure 5.12: output wave form of SAR logic

The SAR ADC's operation is evident through the successive approximation process. The sampled input Vin_samp(blue trace) is equal to 1.29V at 18.7 ns and is compared against the internal DAC's output Vdac(orange trace) reaching 1.30V at 20.6 ns and 592mV at 30.9 ns against a 590mV input. This approximation, guided by the clk and rst signals, is driven by the comp_out to generate the digital code, successfully converting the analog input to digital which has been shown as a bus in the above plots.

Conversion time of SAR-ADC=number of bits of ADC X one bit decision clock period =7X10=70ns

5.5 IMAC 4-bit by 4-bit multiplication Analysis

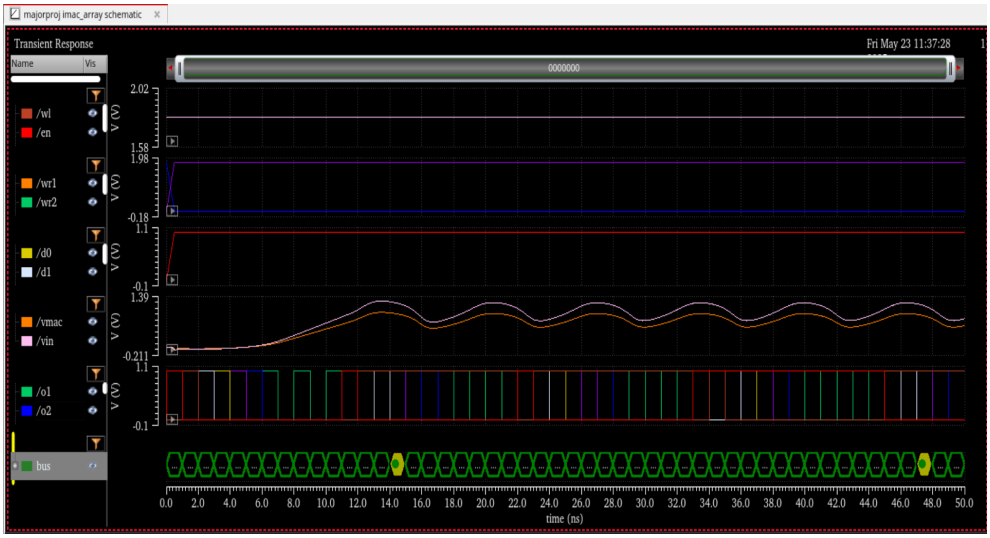

Figure 5.13: Overall Operation Waveforms of the MAC Unit

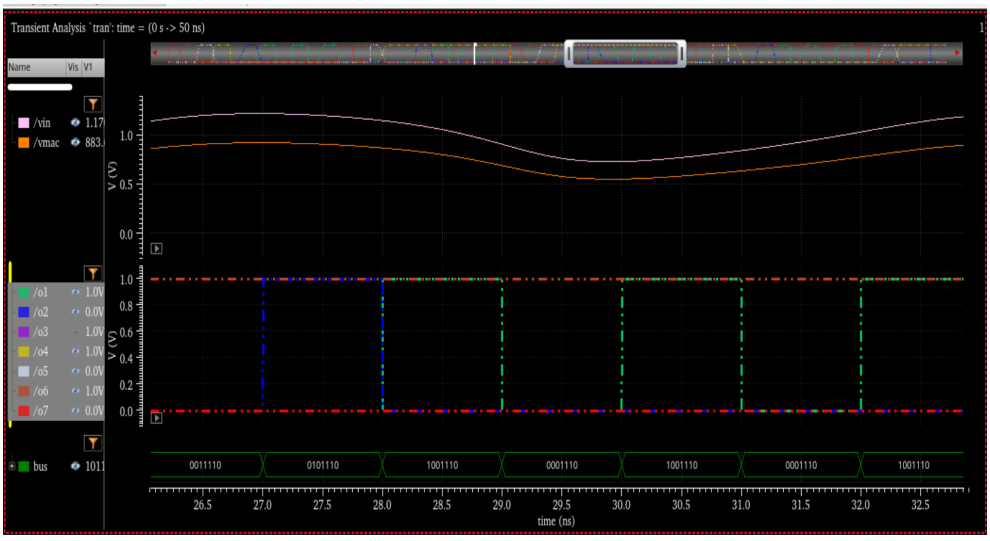

Figure 5.14: zoomed-in view of MAC output and digital bus

waveforms depict the functional operation of a MAC unit, showing the interplay between analog inputs, internal analog processing, and the final multi-bit digital output.

Justification

Time Point : Around 26.5 ns

Given input vector to the 4 Bit DAC=1010

SRAM weights=0011

equivalent analog voltage Vdac=1.125

Obtained Vmac(from the waveforms)=0.429V

SAR ADCt = 0011110 equivalent to 30 (decimal)

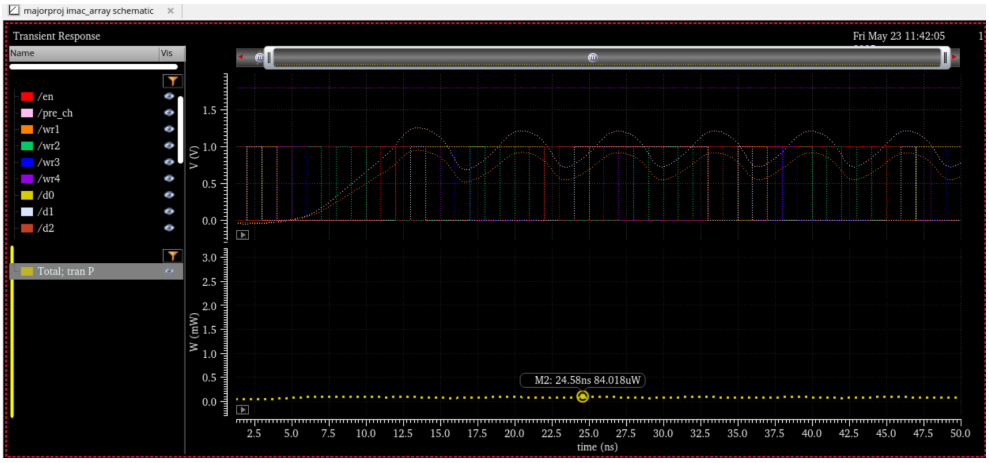

Figure 5.15: power consumption of IMAC

5.6 IMAC 4X4 Array Analysis

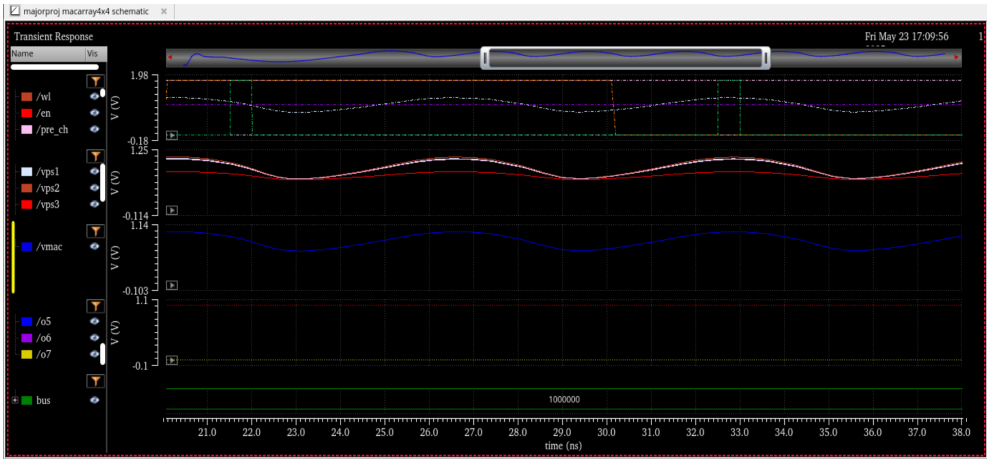

Figure 5.16: Transient analysis waveforms of IMAC Array

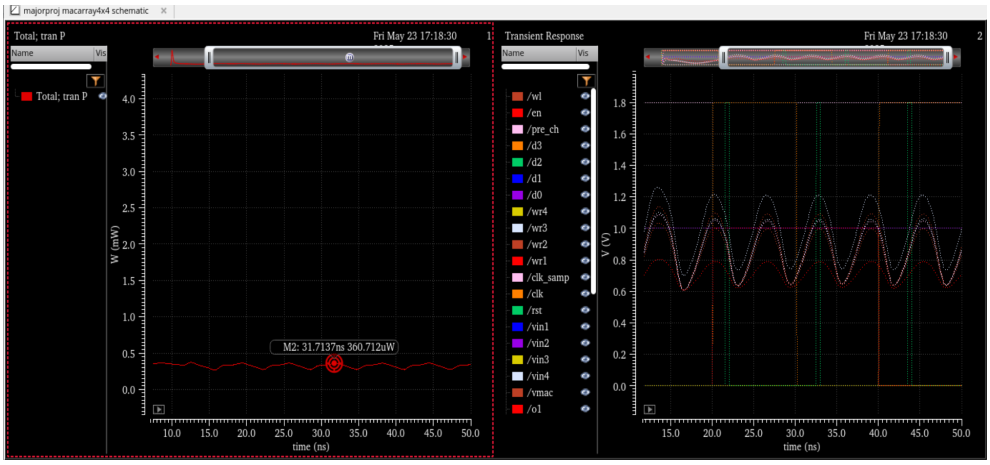

Figure 5.17: Total power consumption of IMAC Array

5.7 Performance Metrics

5.7.1 Timing analysis

SRAM read delay=1.25ns
DAC settling time=17ns
SAR ADC conversion time: 7×10 ns = 70 ns
Total delay=88.25ns

5.7.2 Power consumption

Power consumed by C-DAC=85uW
Power consumed by SAR ADC=1.26mW
Power consumed by IMAC Array=360uW
Total power consumption= 1.705mW

5.7.3 Power Delay Product

PDP= 88.25nS X 1.26 mW = 151 pJ

5.7.4 Throughput

Throughput is the total number of operations that IMAC array can perform per second.

$$\text{Throughput (GOPS)} = \frac{\text{Number of MAC units} \times \text{Clock frequency (Hz)}}{10^9}$$

Number of MAC units=4X4=16
Clock frequency=1GHz
Throughput=16 GOPS

5.7.5 Energy Efficiency

Energy Efficiency is number of operations performed by IMAC per unit of energy.

$$\text{Energy Efficiency (TOPS/W)} = \frac{\text{Throughput (GOPS)}}{\text{Power (W)}}$$

Throughput=16GOPS
Power= 1.705mW
Energy Efficiency =9.378 TOPS/W

5.8 Results of praposed work

Parameter	**This Work**
Technology	180-nm
Bitcell Structure	9T1C
Array Size	4×4
Supply Voltage	1.8 V
Frequency	1GHz
Computing Type	Charge
ADC Type	SAR ADC
Delay ** (ns)	88.25
Power ** (mW)	1.705
Throughput ** (GOPS)	16
Energy Efficiency (TOPS/W)	9.38

Table 5.1: Prelayout simulation results of this work

5.9 Trade-offs and limitations of proposed methodology

Aspect	**Benefit**	**Limitation**
9T1C SRAM	Robust for computation and storage	Occupies more area than 6T SRAM
Charge domain	Energy efficient without active multipliers	Sensitive to charge leakage
Capacitor summation array	Low power because of passive component operation	Area and accuracy purely depends on capacitor matching
SAR ADC	More accuracy	More latency and area overhead

Table 5.2: Benefits and limitations of various components in the proposed IMC architecture

Chapter 6

Conclusion and Future scope

6.1 conclusion

This work presents a 4×4 Multiply-and-Accumulate (MAC) array architecture based on 9T1C SRAM cells, integrated with a charge-domain computational approach and capacitor-based digital-to-analog conversion. The proposed design addresses the limitations of traditional digital MAC units by leveraging in-memory computing to minimize data movement and energy consumption.

6.1.1 Key contributions include:

- The use of 9T1C SRAM cells for simultaneous storage and computation, enhancing robustness and compatibility with IMC.
- A charge-domain computation framework, eliminating the need for power-hungry digital multipliers.
- A capacitor summation array that enables low-power accumulation through passive analog addition.
- Integration with a Successive Approximation Register (SAR) ADC to convert analog accumulated results into digital form with improved accuracy.

6.1.2 Key findings:

- High throughput performance of upto 16 GOPS.
- Low power consumption of 1.705 mW, yielding an energy efficiency of approximately 9.38 TOPS/W

These contributions and metrics highlight the potential of the proposed IMC-based MAC array for energy-efficient and scalable neuromorphic or edge-AI hardware.

6.2 Future scope

Several directions can be pursued to extend and enhance this work in the future:

- **Scaling the array size:** Expanding the architecture to larger arrays such as 16x16 or 32x32 can help evaluate the scalability and computational efficiency of the proposed design.
- **Post-layout simulation:** Performing layout design followed by parasitic extraction and post-layout simulations will provide a more accurate understanding of delay, power, and signal integrity under real-world conditions.
- **Optimization of analog components:** Further refinement of the capacitor summation array and SAR ADC can improve the speed, and power efficiency.
- **Technology scaling:** Exploring implementation in advanced technology nodes such as 65nm or 28nm can offer insights into area, power, and performance benefits for real-world edge AI applications.
- **ASIC tape-out and silicon testing:** Fabricating the design and testing on silicon will help validate the simulation results, identify practical non-idealities, and benchmark the design against existing implementations.

Bibliography

[1] M. Ali, A. Jaiswal, S. Kodge, A. Agrawal, I. Chakraborty, and K. Roy,*IMAC: In-Memory Multi-Bit Multiplication and Accumulation in 6T SRAM Array*, IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 67, no. 8, pp. 2521–2530, Aug. 2020.

[2] S. Lee and Y. Kim,*Charge-Domain Static Random Access Memory-Based In-Memory Computing with Low-Cost Multiply-and-Accumulate Operation and Energy-Efficient 7-Bit Hybrid Analog-to-Digital Converter*, Electronics, vol. 13, no. 3, p. 666, 2024.

[3] Z. Chen et al.,*CAP-RAM: A Charge-Domain In-Memory Computing 6T-SRAM for Accurate and Precision-Programmable CNN Inference*, arXiv preprint arXiv:2107.02388, 2021.

[4] A. Author et al.,*A 9T-SRAM In-Memory Computing Macro for Boolean Logic and Multiply-and-Accumulate Operations*, Microelectronics Journal, vol. 131, 2023.

[5] T. Sakurai et al.,*A 177 TOPS/W Capacitor-Based In-Memory Computing SRAM Macro with Stepwise-Charging/Discharging DACs and Sparsity-Optimized Bitcells for 4-Bit Deep Convolutional Neural Networks*, in IEEE Custom Integrated Circuits Conference (CICC), 2022.

[6] S. Nasrin et al.,*Memory-Immersed Collaborative Digitization for Area-Efficient Compute-in-Memory Deep Learning*, arXiv preprint arXiv:2307.03863, 2023.

[7] A. Author et al.,*A 65nm 4Kb Algorithm-Dependent Computing-in-Memory SRAM Unit-Macro with 2.3 ns and 55.8 TOPS/W Fully Parallel Product-Sum Operation for Binary DNN Edge Processors*, Microelectronics Journal, vol. 131, 2023.

[8] H. Liu et al.,*AFPR-CIM: An Analog-Domain Floating-Point RRAM-Based Compute-In-Memory Architecture with Dynamic Range Adaptive FP-ADC*, arXiv preprint arXiv:2402.13798, 2024.

[9] S. Negi et al.,*HCiM: ADC-Less Hybrid Analog-Digital Compute in Memory Accelerator for Deep Learning Workloads*, arXiv preprint arXiv:2403.13577, 2024.

[10] M. P. E. Apolinario et al.,*Hardware/Software Co-Design with ADC-Less In-Memory Computing Hardware for Spiking Neural Networks*, arXiv preprint arXiv:2211.02167, 2023.

[11] A. Author et al.,*A Fully Digital SRAM-Based Four-Layer In-Memory Computing Unit Achieving Multiplication Operations and Results Store*, IEEE Transactions on VLSI Systems, vol. 31, no. 5, pp. 1053–1063, 2023.

[12] Y. Kim et al.,*Efficient In-Memory Computing With Digitally Controlled Charge-Domain MAC and Low-Power ADC*, IEEE JSSC, vol. 56, no. 10, pp. 3082–3094, 2021.

[13] A. Shafique et al.,*Energy-Efficient Memory-Centric Computing Architectures for AI Acceleration*, IEEE Access, vol. 10, pp. 760–778, 2022.

[14] B. Murmann,*ADC Performance Survey 1997-2023*, Stanford University, [Online]. Available: https://web.stanford.edu/ murmann/adcsurvey.html

[15] H. Jiang et al.,*A 4-bit 2GS/s SAR ADC in 28nm CMOS for Low-Power Applications*, IEEE ISSCC, 2023.

[16] S. Lee and Y. Kim,*A 9T1C SRAM Bitcell for Charge-Domain In-Memory MAC Operations*, in IEEE International Symposium on Circuits and Systems (ISCAS), 2024.

[17] S. Lee and Y. Kim, *Charge-Domain Static Random Access Memory-Based In-Memory Computing with Low-Cost Multiply-and-Accumulate Operation and Energy-Efficient 7-Bit Hybrid Analog-to-Digital Converter*, IEEE Journal of Solid-State Circuits, vol. 56, no. 9, pp. 2675–2686, 2021.